Image Acquisition, Recognition & Speech Conversion

A thesis
submitted towards the partial fulfillment of
the requirements of the degree of

Master of Engineering
in
Electronic Instrumentation and Control Engineering

Submitted By:

Devendra Kumar Somwanshi

Roll No-80751008

Under the esteemed guidance of

Mrs. Gagandeep Kaur
Lecturer (S.G.), EIED
Thapar University, Patiala

ELECTRICAL & INSTRUMENTATION ENGINEERING

DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July - 2009

Acknowledgement

Though it may appear that the following exposition is a monotonous beat of an unusual acknowledgement, I assert beyond the confines of the simple sense of word GRATITUDE. I take it as a highly esteemed privilege in expressing my sincere gratitude to my thesis supervisor Mrs. Gagandeep Kaur, Lecturer (S.G.) for their kind and consistent guidance, encouragement and critical appraisal of the manuscript during the course of this thesis.

I am grateful to Dr. Smarajit Ghosh, Professor and Head in Electrical & Instrumentation Engineering Department for giving me, an encouragement to work on the power factor controller based problem of my interest and providing me his kind co-operation in enriching me in various roles and providing me all necessary facilities to work in the lab.

I am also grateful to Dr. R. K. Sharma, Dean of Academic Affair for his constant encouragement that was of great importance in the completion of the thesis.

I extend my thanks to Dr. Abhijit Mukherjee, Director, Thapar University for his valuable support that made me consistent performer.

Last but not the least: I owe my gratitude to the Almighty for giving me the courage and insight to complete this work. Especially, I would like to give my special thanks to my parents and family, for their patient love and encouragement enabled me to complete this work. My heart-felt thanks to all my colleagues and friends for their help without which this work would not have been successful.

Place: Thapar University, Patiala Devendra Kumar Somwanshi

Date:

CERTIFICATE

This is to certify that my work presented in this thesis entitled **"Image Acquisition, Recognition & Speech Conversion"** submitted in partial fulfillment of the requirement for the award of the degree of **Master of Engineering in Electronic Instrumentation and Control Engineering** at **Thapar University, Patiala**, is an authentic record of my own work under supervision and guidance of **Mrs. Gagandeep Kaur** lecturer (S.G.), Electrical & Instrumentation Department, Thapar University, Patiala. The matter embodied in this report has not been submitted anywhere for the award of any degree.

Date: 15/7/9

Devendra Kumar

Devendra Kumar Somwanshi

Roll No - 80751008

It is certified that the above statement made by the student is correct to the best of our knowledge and belief.

Gagandeep

(Mrs. Gagandeep Kaur)

Lecturer (S.G.), EIED

Supervisor

Thapar University, Patiala

S. Ghosh 15/7/09

(Dr. Smarajit Ghosh)

Professor & Head, EIED

Thapar University, Patiala

RK Sharma 21/7

(Dr. R. K. Sharma)

Dean of Academic Affairs

Thapar University, Patiala

ABSTRACT

The present growth of digitization of books and manuscripts demands an immediate solution to access them electronically. In the last three decades significant advancement is made in the recognition of documents written in Latin-based scripts. There are many excellent attempts in building robust document analysis systems in industry, academia and research labs. While in text to speech, there are many systems which convert normal language text in to speech. This thesis aims to study on image recognition technology (Optical Character Recognition) with speech synthesis technology and to develop a cost effective user friendly image to speech conversion system using MATLAB. In this work we tried to make a system by which we can get the text through image and then speech through that text using MATLAB. The primary motivations are to provide users with a friendly vocal interface with the computer and to allow people with certain handicaps (such as blindness, dumbness, poor vision, visual dyslexia) to use the computer or to read any type of documents. The tremendous recent developments in speech and computer technology have produced unrestricted-vocabulary speech synthesis on PCs in English and some other European languages.

Table of content

List of Figure

List of Abbreviation

Sr. no.	Short form	Abbreviation
1	ANN	Artificial Neural Network
2	DLI	Digital Library of India
3	DSP	Digital Signal Processing
4	HMM	Hidden Markov Model
5	ICR	Intelligent Character Recognition
6	IOCRED	Intelligent Optical Character Recognition Editor
7	IMRC	Intelligent Machines Research Corporation
8	LPC	Linear Prediction Coding
9	LTS	Letter to Sound
10	MICR	Magnetic Ink Character Recognition
11	MODI	Microsoft Office Document Imaging
12	NLP	Natural Language Processing
13	OCR	Optical Character Recognition
14	OHR	Optical Handwriting Recognition
15	OMR	Optical Mark Recognition
16	RMSC	Regional Mega Scanning Center
17	SNN	Syntactic Neural Network
18	TTS	Text To Speech
19	TD-PSOLA	Time-Domain Pitch-Synchronous-OveraLap-Add

Chapter 1

Introduction

1.1 Introduction to OCR

OCR is the acronym for Optical Character Recognition. This technology allows a machine to automatically recognize characters through an optical mechanism. Human beings recognize many objects in this manner our eyes are the "optical mechanism." But while the brain "sees" the input, the ability to comprehend these signals varies in each person according to many factors. By reviewing these variables, we can understand the challenges faced by the technologist developing an OCR system.

First, if we read a page in a language other than our own, we may recognize the various characters, but be unable to recognize words. However, on the same page, we are usually able to interpret numerical statements - the symbols for numbers are universally used. This explains why many OCR systems recognize numbers only, while relatively few understand the full alphanumeric character range.

Second, there is similarity between many numerical and alphabetical symbol shapes. For example, while examining a string of characters combining letters and numbers, there is very little visible difference between a capital letter "O" and the numeral "0." As humans, we can re-read the sentence or entire paragraph to help us determine the accurate meaning. This procedure, however, is much more difficult for a machine.

Third, we rely on contrast to help us recognize characters. We may find it very difficult to read text which appears against a very dark background, or is printed over other words or graphics. Again, programming a system to interpret only the relevant data and disregard the rest is a difficult task for OCR engineers.

1.2 History of Optical Character Recognition

Optical character recognition (OCR) is the process of translating scanned images of typewritten text into machine-editable information. The engineering attempts at automated recognition of printed characters started prior to World War II. But it was not until the early 1950's that a commercial venture was identified that justified necessary funding for research and development of the technology. In the early 1950s, David Shepard was issued U.S. Patent Number 2,663,758 for "Gismo," the first machine to convert printed material into machine language. Shepard then founded Intelligent Machines Research Corporation (IMRC), which produced the first OCR systems for commercial operation. Reader's Digest installed the first commercial system in 1955. The United States Postal Service has been using OCR machines to sort mail since 1965.

The "eye" of early OCR equipment utilized lights, mirrors, fixed slits for the reflected light to pass through, and a moving disk with additional slits. The reflected image was broken into discrete bits of black and white data, presented to a photo-multiplier tube, and converted to electronic bits.

The "brain's" logic required the presence or absence of "black" or "white" data bits at prescribed intervals. This allowed it to recognize a very limited, specially designed character set. To accomplish this, the units required sophisticated transports for documents to be processed. The documents were required to run at a consistent speed and the printed data had to occur in a fixed location on each and every form.

The next generation of equipment, introduced in the mid to late 1960's, used a cathode ray tube, a pencil of light, and photo-multipliers in a technique called "curve following". These systems offered more flexibility in both the location of the data and the font or design of the characters that could be read. It was this technique that introduced the concept that handwritten characters could be automatically read, particularly if certain constraints were utilized. This technology also introduced the concept of blue, non-reading inks as the system was sensitive to the ultraviolet spectrum.

The third generation of recognition devices, introduced in the early 1970's, consisted of

photo-diode arrays. These tiny little sensors were aligned in an array so the reflected image of a document would pass by at a prescribed speed. These devices were most sensitive in the infra-red portion of the visual spectrum so "red" inks were used as non-reading inks. That brings us to this generation of hardware.

Today, OCR technology incorporates high-speed scanners and complex computer algorithms to increase speed and data accuracy. OCR systems no longer require training to read a specific font. Current systems can recognize most fonts with a high degree of accuracy and some are capable of outputting formatted text that closely approximates the printed page.

1.3 Types of recognition engines

1.3.1Optical Character Recognition (OCR)

OCR engines turn images of machine-printed characters into machine-readable characters. Images of machine-printed characters are extracted from a bitmap. Forms can be scanned through an imaging scanner, faxed, or computer generated to produce the bitmap. OCR is less accurate than optical mark recognition but more accurate than intelligent character recognition.

Fig 1.1 Optical character recognition instrument

1.3.2 Intelligent Character Recognition (ICR)

ICR reads images of hand-printed characters (not cursive) and converts them into machine-readable characters. Images of hand-printed characters are extracted from a bitmap of the scanned image. ICR recognition of numeric characters is much more accurate than the recognition of letters. ICR is less accurate than OMR and requires some editing and verification. However, proven form design methods outlined later in this paper can minimize ICR errors.

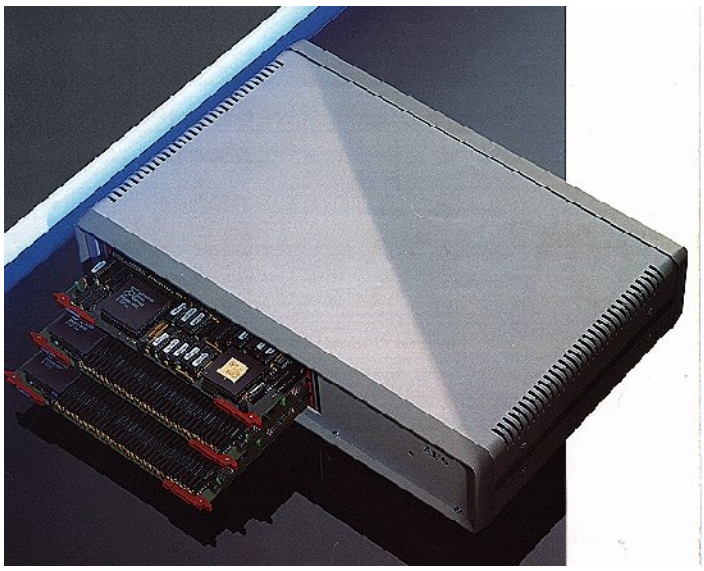


Fig 1.2: Intelligent character recognition instrument

1.3.3 Optical Mark Recognition (OMR)

OMR technology detects the existence of a mark, not its shape. OMR forms usually contain small ovals, referred to as 'bubbles,' or check boxes that the respondent fills in. OMR cannot recognize alphabetic or numeric characters. OMR is the fastest and most accurate of the data collection technologies. It is also relatively user-friendly. The accuracy of OMR is a result of precise measurement of the darkness of a mark, and the sophisticated mark discrimination algorithms for determining whether what is detected is

an erasure or a mark.

Fig 1.3: Optical mark reader

1.3.4 Magnetic Ink Character Recognition (MICR)

Fig 1.4: Magnetic Ink Character Recognition

MICR is a specialized character recognition technology adopted by the U.S. banking industry to facilitate check processing. Almost all U.S. and U.K. checks include MICR characters at the bottom of the paper in a font known as E-13B. Many modern

recognition engines can recognize E-13B fonts that are not printed with magnetic ink. However, since background designs can interfere with optical recognition, the banking industry uses magnetic ink on checks to ensure accuracy.

1.3.5 Barcode Recognition

A barcode is a machine-readable representation of information. Barcodes can be read by optical scanners called barcode readers or scanned from an image using software. A 2D barcode is similar to a linear, one-dimensional barcode, but has more data representation capability.

Fig 1.5: Bar recognition instruments

1.4 Introduction to Speech synthesis

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diaphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice, and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer. Many computer operating systems have included speech synthesizers since the early 1980s.

1.4.1Overview of text processing

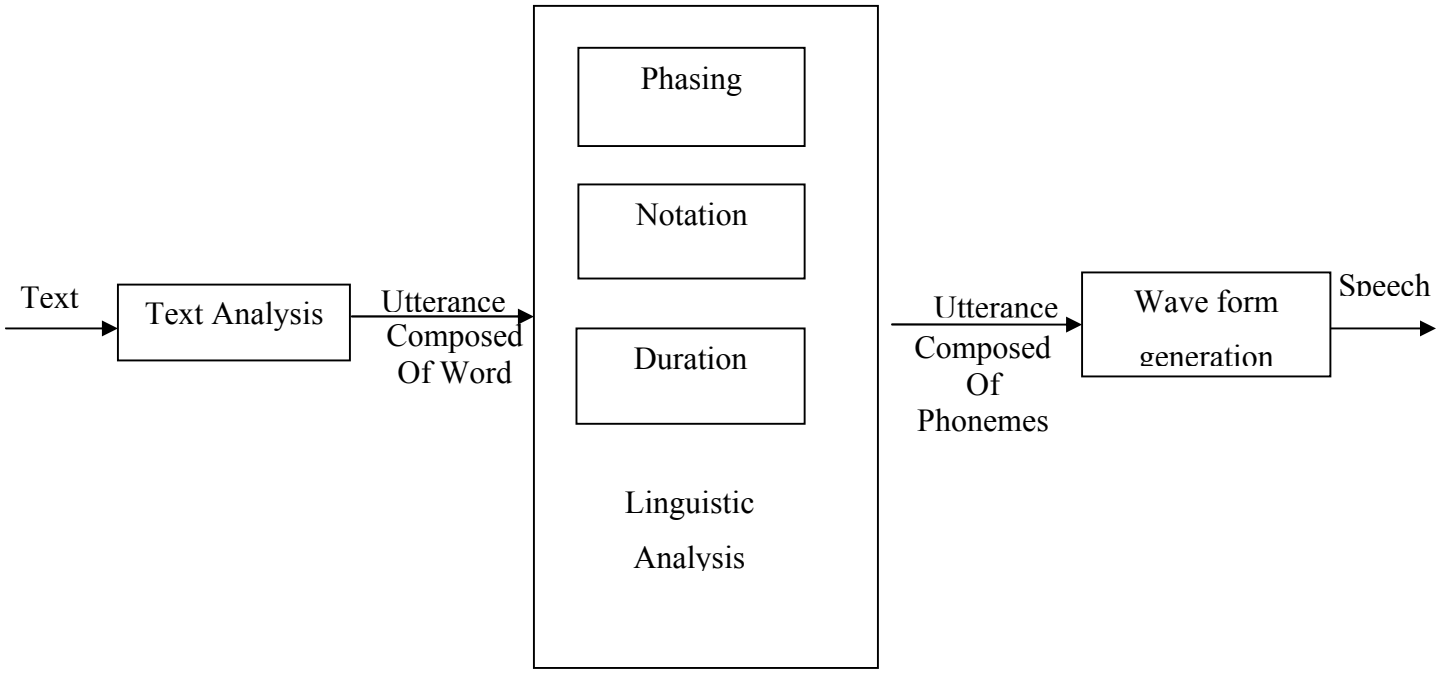


Fig 1.6: Text processing

A text-to-speech system (or "engine") is composed of two parts: a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like

numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end often referred to as the synthesizer, then converts the symbolic linguistic representation into sound.

1.4.2 History of TTS

Long before electronic signal processing was invented, there were those who tried to build machines to create human speech. Some early legends of the existence of "speaking heads" involved Gerbert of Aurillac (d. 1003 AD), Albertus Magnus (1198–1280), and Roger Bacon (1214–1294).

In 1779, the Danish scientist Christian Kratzenstein, working at the Russian Academy of Sciences, built models of the human vocal tract that could produce the five long vowel sounds (in International Phonetic Alphabet notation, they are [a], [e], [i], [o] and [u]. This was followed by the bellows-operated "acoustic-mechanical speech machine" by Wolfgang von Kempelen of Vienna, Austria, described in a 1791 paper. This machine added models of the tongue and lips, enabling it to produce consonants as well as vowels. In 1837, Charles Wheatstone produced a "speaking machine" based on von Kempelen's design, and in 1857, M. Faber built the "Euphonia". Wheatstone's design was resurrected in 1923 by Paget.

In the 1930s, Bell Labs developed the VOCODER, a keyboard-operated electronic speech analyzer and synthesizer that was said to be clearly intelligible. Homer Dudley refined this device into the VODER, which he exhibited at the 1939 New York World's Fair.

The Pattern playback was built by Dr. Franklin S. Cooper and his colleagues at Haskins Laboratories in the late 1940s and completed in 1950. There were several different

versions of this hardware device but only one currently survives. The machine converts pictures of the acoustic patterns of speech in the form of a spectrogram back into sound. Using this device, Alvin Liberman and colleagues were able to discover acoustic cues for the perception of phonetic segments (consonants and vowels).

Early electronic speech synthesizers sounded robotic and were often barely intelligible. The quality of synthesized speech has steadily improved, but output from contemporary speech synthesis systems is still clearly distinguishable from actual human speech.

1.4.2.1 Electronics devices

The first computer-based speech synthesis systems were created in the late 1950s, and the first complete text-to-speech system was completed in 1968. In 1961, physicist John Larry Kelly, Jr and colleague Louis Gerstman used an IBM 704 computer to synthesize speech, an event among the most prominent in the history of Bell Labs. Kelly's voice recorder synthesizer (vocoder) recreated the song "Daisy Bell", with musical accompaniment from Max Mathews. Coincidentally, Arthur C. Clarke was visiting his friend and colleague John Pierce at the Bell Labs Murray Hill facility. Clarke was so impressed by the demonstration that he used it in the climactic scene of his screenplay for his novel 2001: A Space Odyssey, where the HAL 9000 computer sings the same song as it is being put to sleep by astronaut Dave Bowman. Despite the success of purely electronic speech synthesis, research is still being conducted into mechanical speech synthesizers.

1.4.3 Synthesizer technologies

The most important qualities of a speech synthesis system are naturalness and intelligibility. Naturalness describes how closely the output sounds like human speech, while intelligibility is the ease with which the output is understood. The ideal speech synthesizer is both natural and intelligible. Speech synthesis systems usually try to maximize both characteristics.

Fig 1.7: Speech synthesizer

The two primary technologies for generating synthetic speech waveforms are concatenative synthesis and formant synthesis. Each technology has strengths and weaknesses, and the intended uses of a synthesis system will typically determine which approach is used.

1.4.3.1 Concatenative synthesis

Concatenative synthesis is based on the concatenation (or stringing together) of segments of recorded speech. Generally, concatenative synthesis produces the most natural-sounding synthesized speech. However, differences between natural variations in speech and the nature of the automated techniques for segmenting the waveforms sometimes result in audible glitches in the output. There are three main sub-types of concatenative synthesis.

- Diphone synthesis: Diphone synthesis uses a minimal speech database containing all the diphones (sound-to-sound transitions) occurring in a language. The number of diphones depends on the phonotactics of the language: for example, Spanish has about 800 diphones and German about 2500. In diphone synthesis, only one example of each diphone is contained in the speech database. At runtime, the target prosody of a sentence is superimposed on these minimal units by means of digital signal processing techniques

such as linear predictive coding, PSOLA or MBROLA. The quality of the resulting speech is generally worse than that of unit-selection systems, but more natural-sounding than the output of formant synthesizers. Diphone synthesis suffers from the sonic glitches of concatenative synthesis and the robotic-sounding nature of formant synthesis, and has few of the advantages of either approach other than small size. As such, its use in commercial applications is declining, although it continues to be used in research because there are a number of freely available software implementations.

- **Unit selection synthesis:** Unit selection synthesis uses large databases of recorded speech. During database creation, each recorded utterance is segmented into some or all of the following: individual phones, diphones, half-phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a "forced alignment" mode with some manual correction afterward, using visual representations such as the waveform and spectrogram. An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighboring phones. At runtime, the desired target utterance is created by determining the best chain of candidate units from the database (unit selection). This process is typically achieved using a specially weighted decision tree.

Unit selection provides the greatest naturalness, because it applies only a small amount of digital signal processing (DSP) to the recorded speech. DSP often makes recorded speech sound less natural, although some systems use a small amount of signal processing at the point of concatenation to smooth the waveform. The output from the best unit-selection systems is often indistinguishable from real human voices, especially in contexts for which the TTS system has been tuned. However, maximum naturalness typically require unit-selection speech databases to be very large, in some systems ranging into the gigabytes of recorded data, representing dozens of hours of speech. Also, unit selection algorithms have been known to select segments from a place that results in less than ideal synthesis (e.g. minor words become unclear) even when a better choice exists in the database.

➢ **Domain-specific synthesis:** Domain-specific synthesis concatenates prerecorded words and phrases to create complete utterances. It is used in applications where the variety of texts the system will output is limited to a particular domain, like transit schedule announcements or weather reports. The technology is very simple to implement, and has been in commercial use for a long time, in devices like talking clocks and calculators. The level of naturalness of these systems can be very high because the variety of sentence types is limited, and they closely match the prosody and intonation of the original recordings.

Because these systems are limited by the words and phrases in their databases, they are not general-purpose and can only synthesize the combinations of words and phrases with which they have been preprogrammed. The blending of words within naturally spoken language however can still cause problems unless the many variations are taken into account. For example, in non-rhotic dialects of English the "r" in words like "clear" is usually only pronounced when the following word has a vowel as its first letter. Likewise in French, many final consonants become no longer silent if followed by a word that begins with a vowel, an effect called liaison. This alternation cannot be reproduced by a simple word-concatenation system, which would require additional complexity to be context-sensitive.

1.4.3.2 Formant synthesis

Formant synthesis does not use human speech samples at runtime. Instead, the synthesized speech output is created using an acoustic model. Parameters such as fundamental frequency, voicing, and noise levels are varied over time to create a waveform of artificial speech. This method is sometimes called rules-based synthesis; however, many concatenative systems also have rules-based components.

Many systems based on formant synthesis technology generate artificial, robotic-sounding speech that would never be mistaken for human speech. However, maximum naturalness is not always the goal of a speech synthesis system, and formant synthesis systems have advantages over concatenative systems. Formant-synthesized speech can be

reliably intelligible, even at very high speeds, avoiding the acoustic glitches that commonly plague concatenative systems. High-speed synthesized speech is used by the visually impaired to quickly navigate computers using a screen reader. Formant synthesizers are usually smaller programs than concatenative systems because they do not have a database of speech samples. They can therefore be used in embedded systems, where memory and microprocessor power are especially limited. Because formant-based systems have complete control of all aspects of the output speech, a wide variety of prosodies and intonations can be output, conveying not just questions and statements, but a variety of emotions and tones of voice.

Examples of non-real-time but highly accurate intonation control in formant synthesis include the work done in the late 1970s for the Texas Instruments toy Speak & Spell, and in the early 1980s Sega arcade machines. Creating proper intonation for these projects was painstaking, and the results have yet to be matched by real-time text-to-speech interfaces.

- **Articulatory synthesis:** Articulatory synthesis refers to computational techniques for synthesizing speech based on models of the human vocal tract and the articulation processes occurring there. The first articulatory synthesizer regularly used for laboratory experiments was developed at Haskins Laboratories in the mid-1970s by Philip Rubin, Tom Baer, and Paul Mermelstein. This synthesizer, known as ASY, was based on vocal tract models developed at Bell Laboratories in the 1960s and 1970s by Paul Mermelstein, Cecil Coker, and colleagues.

Until recently, articulatory synthesis models have not been incorporated into commercial speech synthesis systems. A notable exception is the NeXT-based system originally developed and marketed by Trillium Sound Research, a spin-off company of the University of Calgary, where much of the original research was conducted. Following the demise of the various incarnations of NeXT (started by Steve Jobs in the late 1980s and merged with Apple Computer in 1997), the Trillium software was published under the GNU General Public License, with work continuing as gnuspeech.

Chapter 2

Literature Survey

2.1 Related work in OCR

R. R. Malyan, S. Sunthankar et.al 1989 have proposed a method to emulate human reading performance. They developed a generalized noise model which used noise to represent the difference between the ideal internal world model consisting of character archetypes and dynamic lexicon, and the actual internal representation [5].

M.R. Lynch and P.J. Rayner 1989 developed a new connectionist model for optical character recognition. They developed a system for recognizing hand drawn capital letter, which is designed to be position, scale and rotation invariant. They investigated the performance of a new unimodal form of connectionist model when coupled to a preprocessing stage. The new connectionist model was briefly derived and its error surface is shown to be unimodal [4].

Shunji Mori, Ching Y. Suen et.al 1992 considered research and development of OCR systems from U historical point of view. They divided their work into two parts: the research and development of OCR systems, and the historical development of commercial OCR's. The R&D part was further divided into two approaches: template matching and structure analysis. It had been shown that both approaches are coming closer and closer to each other and it seemed they tend to merge into one big stream open problems are also raised in their work [7].

Jianli Liu, Nugent et.al 1993 developed a new AI-based OCR post processing technique, implemented as an intelligent OCR Editor (IOCRED), which could enable the automation of OCR post processing procedure and, therefore could result in the increase of throughput, the decrease of error rate and the reduction of cost per page of an OCR system. IOCRED was a novel AI approach to automating OCR post-processing procedures. The IOCRED concept was based on the premise that different OCR

algorithms have distinct error characteristics and such distinction enable a cognitive device to automate its error detection and correction process. IOCRED system should be able to achieve a high throughput, low error rate and low cost OCR conversion. The utilization of the IOCRED technique could result in the removal or the reduction of the current OCR post processing techniques, error rate and cost per page [8].

D.G. Elliman [9] 1994 worked on optical recognition of hand-written documents. The input to this process was a binary image produced by scanning the original document. A grey-level image contains more information, and is a necessary first stage where the original exhibits poor contrast between the written symbols and the background. The original document was binary in character, having the two states ink and no-ink. Thus no significant information was lost in a binary representation given a sufficient resolution.

S.M. Lucas [11] 1995 described the application of a special type of syntactic neural network (SNN) to the recognition of hand-written digits. Importantly, it was shown that this class of SNN can be implemented to work at very high classification speeds (similar to that of an N-tuple classifiers**)** but with higher classification accuracy when trained on enough data. Results were reported on the ESSEX and CEDAR datasets to demonstrate this.

Luis R. Blando, Junichi Kanai et.al 1995 proposed A classifier for predicting the character accuracy achieved by any Optical Character Recognition (OCR) system. This classifier was based on measuring the amount of white speckle, the amount of character fragments, and overall size information in the page. No output from the OCR system is used. The given page was classified as either “good” quality (i.e., high OCR accuracy expected) or “poor” (i.e., low OCR accuracy expected). Results of processing 639 pages show a recognition rate of approximately 85%. This performance compared favorably with the ideal-case performance of a prediction method based upon the number of reject-markers in OCR generated text. They suggested that some OCR errors are not caused by image defects. This method did not detect such errors [10].

C. Tanprasert, T. Koanantakool [16] 1996 proposed Thai Optical Character Recognition. They proposed to apply the artificial neural networks (ANNs) together with some pre-processing and post-processing techniques to solve the Thai OCR problem. The experimental result confirmed that ANNs is a very suitable technique for developing the Thai OCR software. The recognition rate on the real document of training fonts is about 90% - 95%. This leaded to a possible implementation in production-quality OCR software that NECTEC software technology laboratory working on.

Sun-Hawa Hahn et.al 1999 proposed a technique on utilizing OCR technology in building text database. They described the points to be considered when one selects an OCR system in order to build database. Based on their experiments on four commercial OCR systems, they choose one that shows the highest recognition rate to build OCR-text database. The character recognition rate marks 90.5 % over 970 abstracts of conference proceedings in Korean. This recognition rate was still insufficient for practical use. In that research, they applied morpheme-based indexing and 2-gram based indexing to the OCR database. The experimental result showed that the retrieval efficiency of OCR database is moderately worse than that of original database. Therefore, OCR technology can be used to create indexes from the document images. Morpheme based indexing showed better retrieval efficiency for the original database. However, the advantage of the morpheme analysis was no longer effective in the case of the OCR database [21].

Jaehwa Park, Venu Govindaraju et.al 2000 proposed a character recognition methodology that achieves high speed and accuracy by using a multi-resolution and hierarchical feature space. Features at different resolutions, from coarse to fine-grained, are implemented by means of a recursive classification scheme. Typically, recognizers have to balance the use of features at many resolutions (which yields a high accuracy), with the burden on computational resources in terms of storage space and processing time. They introduced a method that adaptively determines the degree of resolution necessary in order to classify an input pattern. This leads to optimal use of computational resources. The Hierarchical OCR dynamically adapts to factors such as the quality of the input pattern, its intrinsic similarities and differences from patterns of other classes it is being compared against, and the processing time available. Furthermore, the finer resolution is accorded to only

certain “zones” of the input pattern which are deemed important given the classes that are being discriminated [22].

J. C. Lecoq, L. Najman et.al 2001 provided benchmarking method for commercial OCR engines with respect to their inclusion in the global digitalization chain from scanning to understanding the text information contained in a technical drawing document. The crucial point was the manual correction of OCR recognition errors. By benchmarking, they intended to identify, for their application domain, the causes for OCR errors which are the most costly to correct. For a given OCR engine, they modeled the correction cost as a function of image characteristics. Their methodology relied on the following issues: i) The design of the correction cost, representing the difficulty of correction for a human operator. ii) The classification of image characteristics that may lead to OCR recognition errors. This methodology allowed us to obtain a list of domain- dependant problems for OCR engines, classified by importance with respect to the correction cost. This listcould then be used to correctly choose the OCR engine, or to enhance the OCR execution, by focusing on the most important problems [24].

Yasuto Ishitani [23] 2001 has proposed model based information extraction method tolerant of OCR errors for document images as the basis for a document reader which can extract required keywords and their logical relationship from various printed documents. Such documents obtained from OCR results may have not only unknown words and compound words, but also incorrect words due to OCR errors. To deal successfully with OCR errors, they adopted robust keyword matching which searches for a string pattern from two dimensional OCR results consisting of a set of possible character candidates. This keyword matching used a keyword dictionary that includes incorrect words with typical OCR errors and segments of words to deal with the above difficulties. After keyword matching, a global document matching was carried out between keyword matching results in an input document and document models which consist of keyword models and their logical relationship. This global matching determined the most suitable model for the input document and solved word segmentation problems accurately even if the document has unknown words, compound words, or incorrect words.

Dave Desrochers, Zhihua Qu et.al 2001 developed the algorithms for testing partially damaged characters. Ever since the character strings on silicon wafers had been read using OCR cameras, there had been a problem with damaged characters. This problem was due to reflection from the light source or the physical damage of the characters themselves. There were obvious types of damage that occur frequently on many of the bitmaps that the OCR camera reads. With these types, one can test them to find the most damaging types on each particular character that has occurred. However, currently there is no known research that systematically determines the worst damages or limits of damage to characters for specific OCR methods such as template matching or neural network algorithms. They gave algorithms for testing common forms of damages on template-matching optical readers reading strings on silicon wafers. It also displayed results from combining a simple neural network and the algorithms. The results on readability study were critical for the development of robust OCR systems. They introduced several methods of studying the effects of damages on OCR algorithms and, more importantly, determined the limits of damages in percentage before characters become unreadable [25].

George Nagy, Prateek Sarkar 2004 have proposed four methods of converting paper documents to computer readable form were compared with regard to hypothetical labor cost: keyboarding, omnifont OCR, style specific OCR, and style constrained or style adaptive OCR. According to them the best choice was to determined primarily by i) the reject rates of various OCR systems at a given error rate, ii) the fraction of the material that must be labeled for training the system, and iii) the cost of partitioning the material according to style [27].

M. Sarfraz, A. Zidouri et.al 2005 have proposed a novel approach for skew estimation of document images in OCR system. They proposed multi-scale properties of an image are utilized together with Principal Component Analysis to estimate the orientation of principal axis of clustered data. The proposed scheme, utilizes multi-scale analysis of an image to detect the tilted angle. We have utilized 'Haar' wavelet to decompose image into detail sub images at various level. In each level, they applied PCA to estimate the orientation of principal axis in horizontal, vertical and diagonal details co-efficient. This

scheme had been tested extensively on Arabic fonts which are connecting in nature and English fonts which are isolated in nature. Their proposed scheme was accurate with in all practical limits for both font systems [28].

Shaolei Feng and R. Manmatha 2006 have proposed a hierarchical, HMM- based automatic evaluation of OCR accuracy for digital library of books. They proposed a Hidden Markov Model (HMM) based hierarchical alignment algorithm to align OCR output and the ground truth for books. They believed this was the first work to automatically align a whole book without using any book structure information. The alignment process worked by breaking up the problem of aligning two long sequences into the problem of aligning many smaller subsequences. This can be rapidly and effectively done. Experimental results showed that their hierarchical alignment approach works very well even if OCR output has a high recognition error rate. Finally, they evaluate the performance of a commercial OCR engine over a large dataset of books based on the alignment results [29].

2.2 Related work in Text to Speech

William A. Ainsworth 1973 proposed a system for converting English text to speech. He investigated the feasibility of converting English text into speech using an inexpensive computer and a small amount of stored data. He segmented text into breath groups, the orthography is converted into a phonemic representation, lexical stress is assigned to appropriate syllables, then the resulting string of symbols in converted by synthesis by rule in the parameter values for controlling an analogue speech synthesizer. The algorithms for performing these conversions are described in details and evaluated independently, and the intelligibility of the resulting synthetic speech in assed by listening tests. In his approach a typical seven word sentence contains less than one phonetic error. This level of performance is probably achieved because most of the longer words in English are pronounced according to rule, whereas the common words are pronounced irregularly. The present set of rules ensures that the most common irregular words are treated as special cases, and the correct phonemic translation is generated [1].

Fushikida, Katsunobu et.al 1982 developed a low cost, compact text to speech synthesizer using formant speech synthesizer LSI for the personal computer. This speech synthesis system based on synthesis by rule using glottal pole formant model and CV, VC speech segment compilation technique. It can synthesize any arbitrary Japanese speech from the input text. They obtained 99% word intelligibility in the listening tests. It should be possible to apply this system for other languages, by modifying the synthesis program and parameters [2].

Susan R. Hertz 1986 proposed English text to speech conversion with delta. Delta is a computer system for developing text to speech rules for any language. He described a set of delta rules for English, on the basis of the input text. The English rules build multi-level linguistic constituents of the utterance (e.g. morphs, syllables, and phonemes) and the low-level synthesizer parameter patterns. In his proposed rules syllable and phoneme tokens are generated and used them to derive target and transitioned patterns for a formant synthesizer, another rule set might take a different approach, perhaps generating diphone tokens, and extracting LPC coefficients for them from the dictionary. Another rule set might use a formant synthesizer, but generate the formant values on the basis of articulatory streams. The possibilities are endless [3].

Cecil H. Coker et.al 1990 proposed two powerful alternatives to letter to sound rules for speech synthesis which are Morphology and Rhyming. Most speech synthesizers have tended to depend on letter-to-sound rules for most words, and resort to a small ‘‘exceptions dictionary’’ of about 5000 words to cover the more serious gaps in the letter-to-sound rules. The Bell Laboratories Text-to-Speech system which is moving to an extreme dictionary-based approach cuts the error rate by at least an order of magnitude. The pronunciation problem has traditionally been divided into two very separate modules: letter-to-sound rules and the exceptions dictionary. They focused on letter-to-sound rules which work from first principles. In contrast, they resort to letter-to-sound rules only when all alternatives have been exhausted. The most reliable inference is table lookup. Failing that, the system tries to make as safe an inference as possible from similar words in the dictionary. Stress neutral morphology is considered fairly safe but rhyming

is more dangerous, but far more this approach has a much smaller error rate than previous letter-to-sound systems.

Mark Tatham and Eric Lewis 1996 proposed that naturalness in synthetic speech is essentially the successful rendering of variability in the final acoustic signal, once they got the obvious factors such as limiting the domain of discourse within which the system is to operate. The obvious factors of rhythm and intonation there is the more difficult question of modeling the variability in human speech. They discussed how SPRUCE, a high-level text- to-speech synthesis system, incorporates several different types of variability. In SPRUCE we identify and treat distinctly several sources of variability in human speech, adhering carefully to contemporary speech production theory. They believed that this approach renders transparent the sources of naturalness, and at the same time enables us to manipulate what they felt to be an important interplay between the various types of variability [15].

Chen Fang and Yuan Baozong 1996 developed the intelligent speech production system with text generation intelligent speech production system is different from ordinary speech production systems. They considered not only how to convert text into speech but also how to generate the necessary text in text-to- speech conversion. The system gets the right test by the step of topic selection, test planning. test organization, grammar realization and text generation According to the generated test the system at last generates speech which have good quality in naturalness and intelligibility using Chinese Text-to-Speech Conversion System. The paper introduced the structure and main techniques of intelligent speech production system with text generation. It has wide application prospect in intelligent demonstration system, expert system, inquiring system, auto report generation system, and translation system to raise the intelligent level in man machine interaction [14].

Nobuyuki Katae and Shinta Kimara 1996 developed natural prosody generation for domain specific text to speech system. In this method, sentences are composed by inlaying a variable word into each slot in prepared sentence structures. This method can

be used for domain specific text-to-speech applications that don't require so many number of sentence structures but many words [12].

Leija L. et.al 1996 developed a system of text reading and translation to voice for blind persons. The developed system falls into text to speech reading machines and can give to the blind, capacity autonomous of reading text once he learned drive it. The read characters are obtained through an optic lector and a computer program based in pattern recognition by means of an artificial neural network. The system used a central processing unit (CPU) from a personal computer PC, containing only a hard disk, a scanner unit, and a sound blaster audio card. The main key in this project was the words recognition by back propagation neural network which is now very useful, and the use of standards cards. Since the first Kurzweil machine (1) text to speech in 1978 with a prize of 30,000 dollars to the modern computer with a storing high capacity and great velocity and the development techniques of pattern recognition with neural networks, today it is possible the development of relatively inexpensive systems, recognizing the importance of the developed algorithms for pattern recognition, which facilitates the development of aids for blind persons. The project was a first step in a text reading machine for any size characters and is planned for basic teaching in public school [13].

Rivarol Vergin, Douglas O'Shoughnessy et.al 1997 proposed an approach to increase the possibilities of speech modifications while preserving most of the speech quality of the original signal. They examined one of the most used methods to modify the current aspect of a speech signal that was the one based on a modification of its apparent pitch contour. The first step of this technique was an estimation of the pitch epochs, followed by a pitch-scale transformation to modify the apparent gender of a current speaker. The apparent rate and intensity contour of the original signal were preserved through the use of a time scale modification. The resulting synthesis signal had an apparent pitch contour relatively different from the input signal. This technique when combined with the pitch-scale and time-scale modification procedure, allowed reaching this goal while preserving most of the speech quality of the original signal [17].

A. P. Breen 1997 suggested that the next generation of synthesis systems will inevitably take more account of the type of information being presented to them, that the interfaces to such systems will become more generic and that the type of processing conducted as part of the synthesis process will become more diffuse and data orientated. He also suggested that advances in speech synthesis can best be achieved when developed within a complete multi-modal spoken language framework [18].

Leija L. et.al 1999 have proposed a reader instrument for translating text to speech using commercial components as a multimedia computer, a flat bed scanner unit, a control and recognition of characters program, the instrument is used to read printed text by blind people. In their proposed work the character recognition is based on a block of pre-classification and artificial neuronal back propagation network, the character are read, through the scanner text from elemental teaching books, and the program locates the words recognized in a abase of recorded words to emit them via voice through the multimedia system. Their system had a limitation; it works with the books of text used in elemental education. The kind of characters is Arial in size of 12 pixels, with a performance of 93% [19].

Mingli Song, Chun Chen et.al 2003 developed 3D Realistic Talking Face Co-Driven by Text and Speech system. The text is translated into a sequence of visemes transcription. And time vector of the sequence is extracted from the speech corresponding to the text after it is segmented into phonetic sequence. A muscle based viseme vector is defined for static viseme. And then, with the time vector and the static visemes's sequence, dynamic visemes are generated through time-related dominance function. Finally, according to the frame rate to be rendered, intermediate frames are interpolated between key frames to make the animation result looks more natural and realistic than those obtained based on the text or speech-driven only.

Soumyajit Dey et.al 2007 proposed architectural optimizations for text to speech synthesis in embedded system. The increasing processing power of embedded devices has created the scope for certain applications that could previously be executed in desktop environments only, to migrate into handheld platforms. An important feature of

the computing systems of modern times is their support for applications that interact with the user by synthesizing natural speech output. In their work, the performance of a Text to Speech Synthesis application is evaluated on embedded processor architectures and medications in the underlying hardware platform are proposed for real time performance improvement of the concerned application [31].

Dmitri Bitouk and Shree K. Nayar 2008 proposed a complete framework for creating a speech enabled avatar from a single image of a person. They used a generic facial motion model which represented deformation of a prototype face during speech. They developed a HMM-based facial animation algorithm which took into account both lexical stress and co-articulation. Their algorithm produced realistic animations of the prototype facial surface from either text to speech. They showed several examples of avatars that were driven by text to speech inputs. Such avatars were animated from text or speech input with the help of a novel motion synthesis algorithm. They developed a method for synthesizing eye gaze motion from a photograph. They demonstrated that their approach can also be used to build volumetric displays that feature speech-enabled 3D avatars [32].

Chapter 3

Optical Character Recognition

3.1 Digital Library

Digital Libraries have received wide attention in the recent years allowing access to digital information from anywhere across the world. They have become widely accepted and even preferred information sources in areas of education, science and even with other information needs. The rapid growth of Internet and the increasing interest in development of digital library related technologies and collections helped to accelerate the digitization of printed documents in the past few years.

A digital library is a library in which collections are stored in digital formats (as opposed to print, microform, or other media) and accessible using computers. The digital content may be stored locally, or accessed remotely via Internet. Digital libraries play a critical role in organizing, preserving and providing access to the various information resources of society. Digital libraries are advantageous as a means of easily and rapidly accessing books, archives and images of various types. Traditional libraries are limited by storage space; digital libraries have the potential to store much more information, simply because digital information requires very little physical space to contain them. Digital libraries can immediately adopt innovations in technology providing users with improvements in electronic and audio book technology. In summary, the following are some of the advantages of having a digital library.

- No physical boundary: The user of a digital library need not to go to the library physically; people from all over the world can gain access to the same information, as long as an Internet connection is available.
- Structured approach: Digital libraries provide access to much richer content in a more structured manner, i.e., we can easily move from the catalog to the particular book then to a particular chapter and so on.
- Information retrieval: The user is able to use any search term (such as word, phrase, title, name, and subject) that are supported by the search engine to search the entire

collection that satisfy their need.

- Preservation and conservation: An exact copy of the original can be made any number of times without any degradation in quality.
- Resource sharing: A particular digital library can provide a link to any other resources of other digital libraries very easily; thus a seamlessly integrated resource sharing can be achieved.
- Multiple accesses: Any number of users can access the same document at the same time. In addition, digital libraries can be used at any time whenever users need information.

With all these advantages of digitization report that, at present, paper documents remain the medium of choice for reading, even when the most high-tech technologies are to hand. They point four principal reasons for this: (i) paper allows flexible navigation through the content; (ii) paper assists cross-referencing of several documents at one time; (iii) paper invites annotation; and (iv) paper allows the interweaving of reading and writing. It is illuminating to bear these considerations in mind when identifying obstacles to the delivery of document images via digital libraries. Of course, efforts are underway to commercialize electronic document displays offering even more of the affordances of paper including flexibility, low weight, low power, and low cost.

3.1.1 Current Status

There are many thousands of digital library projects initiated and funded by Digital Libraries Initiatives in US, Europe, Asian and other part of the world. This is done with the aim of digitizing the important documents, both to preserve them for future generations and to make them easily accessible. These days, large scale digitization projects are underway at Google, the Million Book Project, MSN, Yahoo, etc. With continued improvements in book handling and presentation technologies, such as optical character recognition, eBooks, and development of alternative business models, digital libraries are rapidly growing in popularity as demonstrated by many projects around the world. Just as libraries have venture into audio and video collections, so have digital libraries such as the Internet Archive.

There are many collaborative digitization projects throughout the globe, such as the Collaborative Digitization Project, Open Content Alliance, etc. These projects help to establish and publish best practices for digitization and work with regional partners to digitize cultural heritage materials.

Among others, the leading projects in the field of digital archive creation and management include project Gutenberg, Google Book Search, Windows Live Search Books, Internet Archive, Cornell University, The Library of Congress World Digital Library, The Digital Library at the University of Michigan, CMU's Universal Library, etc.

The Million Book Project (or the Universal Library), led by Carnegie Mellon University achieved the target of digitizing a million books by 2007. The project has been working with government and research partners in India, China, Egypt, etc. As a result of which the Digital Library of India project was initiated in November, 2002.

3.1.2 Digital Library of India

Since its inception in the year 2002, the digital library of India project currently digitizes and preserves books, though one of the future avenues is to preserve existing digital media of different formats like video, audio, etc. The scanning operations and preservation of digital data takes place at different centers across India, Regional Mega Scanning Center (RMSC). The RMSCs themselves function as individual organizations with scanning units established at several locations in the region.

The Digital Library of India (DLI) project aims to digitally preserve all the significant literary, artistic and scientific works of people and make it freely available to anyone, anytime, from any corner of the world, for education, research, etc. The project has been successfully digitizing books, which are a dominant store of knowledge and culture. It hosts close to one third of a million books online with about 70 million pages scanned at almost 30 centers across the country. The scanning centers include academic institutions of high repute, religious and government institutions. Summary of the diverse collections of documents archived in Digital Library of India is presented in Table 3.1.

2.1.3 Challenges

The rapid growth of digital libraries worldwide poses many new challenges for document image analysis research and development. Digital libraries promise to offer more people access to larger document collections, and at far greater speed, than physical libraries can. But digital libraries also tend, for many reasons, to serve poorly, or even to omit entirely, many types of paper-based printed or handwritten documents. These documents, in their original physical (undigitized) form, are readily legible, searchable, and browse able, whereas in the form of document images accessed through digital libraries they often lose many of their original advantages.

Documents	Diversity of Books
Languages	Hindi, Telugu, Urdu, Kannada, Sanskrit, English, Persian, European, others
Typesets	Letter press, Offset printer, Typewriter, Computer Typeset, Handwritten
Fonts	Ranging from 10 to 20 fonts in each language
Styles	Italics, Bold, Sans serifs, Underline, etc.
Sizes	8 to 45 points
Year of Publication	Printed books from 19, 20 and 21st centuries Ancient manuscripts to 2005

Table 3.1: Diversity of document collections in DLI. They vary in languages, fonts, styles, sizes and typesets. Books that are published since 1850 are archived.

This is because existing document image understanding techniques cannot fully give solution to the challenges of digitized document images. The unusual wide variety of document images found in digital libraries, representing many languages, cultures, and historical periods, tend to pose particularly severe challenges for present day digital image analysis systems. These systems are not robust in the face of multilingual text and non-Latin scripts, unusual typefaces, and poor image quality.

The Need for Accurate Transcriptions of Text: The central classical task of digital image analysis research has been, for decades, to extract a full and perfect transcription of the textual content of document images. Although perfect transcriptions have been known to result, no existing OCR technology, whether experimental or commercially available, can guarantee high accuracy across the full range of document images of interest to users. Even worse, it is rarely possible to predict how badly an OCR system will fail on a given document.

Indexing and Retrieval: The indexing and retrieval of document images are critical for the success of digital libraries. Most reported approaches for retrieval of document images first attempt recognition and transcription followed by indexing and search operating on the resulting (often error full) encoded text, using standard “bag-of-words" information retrieval methods. Although for some retrieval tasks, error rates typical of commercial OCR machines do not seriously degrade recall or precision of statistical \bag-of-words" methods, some textual analysis tasks (e.g. depending on syntactic analysis), whether modeled statistically or symbolically, can be derailed by even low OCR error rates.

The attempts made on word spotting for retrieval from document images without explicit representation is encouraging. This approach seems to offer the greatest promise of large improvements in effectiveness of document image retrieval (if not in speed). It needs further investigation to reach the level of text information retrieval methods. We review hereunder some of the e orts exerted in similar directions for developing systems for document image recognition and also for document image retrieval in the image domain.

3.2 Recognition of Document Images

Recognition of document images is at the heart of any document image understanding system. Optical character recognition (OCR) systems take scanned images of paper documents as input, and automatically convert them into digital format for computer-aided data processing. The potential of OCR for data entry application is obvious: it offers a faster, more automated, and presumably less expensive alternative to the manual data entry devices, thereby improving the accuracy and speed in transcribing data into the

computer system. Consequently, it increases efficiency and effectiveness (by reducing cost, time and labor) in information storage and retrieval.

OCRs have wide range of applications in government and business organizations, as well as individual companies and industries. Some of the major applications of OCR include:

(i) Library and office automation, (ii) Form and bank check processing, (iii) Document reader systems for the visually impaired, (iv) Postal automation, and (v) Database and corpus development for language modeling, text-mining and information retrieval, Optical character recognition has a relatively long history. The technology was invented in the early 1800s, when it was patented as reading aids for the blind. In 1870, C. R. Carey patented an image transmission system (the retina scanner) using a mosaic of photocells, and in 1890, P.G. Nipkow invented the sequential scanner, whereby an image was analyzed line by line. This was a major breakthrough for reading machines. However, the practical OCR technology used for reading characters was introduced in the early 1950s as a replacement for keypunching system used in business. A year later, D.H. Shephard developed the first commercial OCR for typewritten data.

It is in 1960 that OCR began to make a noticeable impact on commercial and government data processing installations. The first postal address readers and the social security administration machines to read typewritten documents were installed in 1965. In 1970there emerged the first OCR system capable of reading a wide variety of forms size, from small documents to large pages. The 1980's saw the emergence of OCR systems intended for use with personal computers. An early successful attempt in implementing character recognition as an aid to the visually handicapped was made by Tyurin in 1990. Nowadays, it is common to find PC-based OCR systems that are commercially available. However, most of these systems are developed to work with Latin-based scripts.

OCR can be done offline or online. OCR converts document images to text that are printed or handwritten. Handwriting recognition is often called optical handwriting recognition (OHR) analogous to optical character recognition (OCR). OCR usually deals with the interpretation of offline data which describes printed or typewritten objects. The

goal of OHR is to interpret the contents of the handwritten data and generate a description of that interpretation in the desired format. The OHR task is also referred to as o -line handwriting recognition or online handwriting recognition. On-line handwriting recognition deals with a data stream which is coming from a transducer while the user writes, and o - line handwriting recognition deals with a dataset which has been obtained from a scanned handwritten document. We are concerned in this work on the OCR of printed document images.

3.2.1 Overview of OCR Design

Figure 3.1 shows the framework of OCR. Most of the designs in OCR follow a modification of this architecture. Given a page for recognition, first it is preprocessed. The aim of the preprocessing module is to prepare the image for recognition. Preprocessing involves binarization, skew correction and normalization. It undergoes some image enhancements such as filtering out noise and increasing the contrast. Then, the image is segmented to separate the characters from each other. Segmentation occurs at two levels. On the first level, text, graphics and other parts are separated. On the second level, text lines, words and characters in the image are located. Information from connected component analysis and projection analysis can be used to assist text segmentation. Segmentation is followed by feature extraction, which is concerned with the representation of the object.

Feature extraction and classification are the heart of OCR. The character image is mapped to a higher level by extracting special characteristics and patterns of the image in the feature extraction phase. Feature extraction is expected to make the image invariant to rotation, translation, scaling, line-thickness, etc. It could also remove redundant information to compress the data amount and a lot of other things. A number of global and local features have been employed for this, including pro les, moments, structural features, discrete Fourier descriptors, etc.

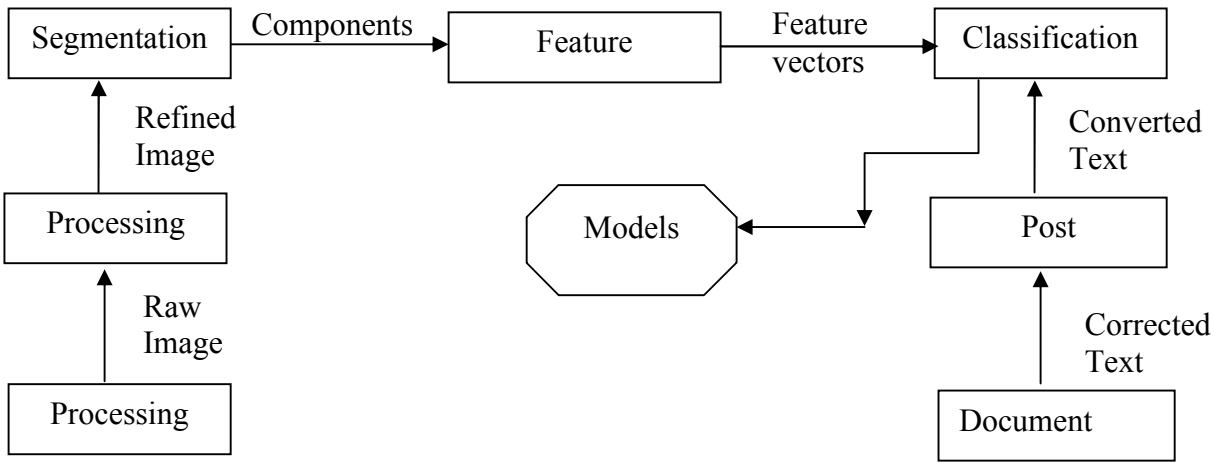


Fig 3.1: Overview of conventional approach of OCR.

The classifier is then trained with the extracted features for classification task. The classification stage identifies each input character image by considering the detected features. A range of classifiers are in use for these purpose. Classifiers such as Template Matching, Neural Networks, Syntactical Analysis, Hidden Markov Models, Bayesian theory, SVM, etc. have been explored. For improving the recognition result, post-processing module is incorporated. The post-processor is typically intended to improve accuracy by detection and correction of OCR errors. Contextual information such as character n-gram, dictionary, semantic knowledge and domain knowledge are exploited in post processing. Below we review some of the notable OCR systems developed in industry, academia and research institutions.

3.2.2 Commercial and Free OCR Systems

Recognition of scanned document images using OCR is now generally considered to be a solved problem for some scripts. This is because high accuracy OCR systems are reported for some languages in the world that use Latin and non-Latin scripts.

Many commercial packages are available for Latin scripts. Some of them are the following: Omni Page Pro from Caere, Fine Reader from ABBYY, Text Bridge from Xerox, Capture from Adobe, etc. These packages do an impressive job on high quality original documents with accuracy rates in the high nineties. There are also sophisticated OCRs for languages

such as Chinese, Japanese and Arabic scripts. A commercial OCR software, 'Presto! DanChing' from NewSoft performs OCR on Chinese, Japanese and Roman characters. For Arabic language the pioneer 'Sakhrs' automatic reader is available commercially that supports also English, French and some other languages. Recently, Caere announces also that, in addition to English and thirteen Western European languages, it provides the first Omnifont OCR software for the Arabic language.

There are also commercial products for postal address reading and a reading machine for a blind or visually impaired. The architecture of the postal address reading machine is designed to achieve correct interpretation of text, as well as high speed in performing the interpretation. The reading machine finds and interprets addresses on a stream of postal letters. The addresses can be either machine-printed or handwritten. Here, the primary subtasks correspond to finding the block of text corresponding to the destination address, recognizing characters and words within the address, and interpreting the text using postal directories. An integrated OCR with speech output system for the blind has been marketed by Xerox and others for English language. The reading machine includes a user input device including a user activated button that produces a signal that causes the computer to enter a definition mode to read a definition of current word being spoken by the reading machine.

Unlike commercial OCRs, there are few free and open source OCRs packages available. Some of these are (i) GOCR (or JOCR) from source forge (an open source OCRs program), (ii) Orad from GNU (an open source OCRs program), (iii) Tesseract OCR from Google (a free OCR engine), etc. Microsoft Office application also contains Microsoft Office Document Imaging (MODI) that supports editing documents scanned by Microsoft Office Document Scanning. It was first introduced in Microsoft Office XP and is included in later Office versions including Office 2003 and Office 2007. Among the many functionality of MODI is it allows users to scan single or multi-page documents and produce editable text from a scanned document using OCR. Though OCR systems developed by commercial vendors give high recognition results on good quality pages, evaluation of current OCRs show that the diversity of document collections (language-wise, quality-wise, time-wise, etc.) reduce the performance of OCR systems greatly. The creation of new fonts, the

occasional presence of decorative or unusual fonts, and degradations caused by faxed, aged and multiple generation of copies, continues to make existing OCRs a topic of research for better accuracy and reliability. This is because most of the present systems have been centered on developing fully automatic systems targeted toward the recognition of a single page at a time. Recent evaluation of commercial software by Lin, argue that document recognition research is still in great need for better accuracy and reliability, complementary contents, or downstream information retrieval.

Chapter 4

Text to speech

4.1 Introduction

A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read any text aloud, whether it was directly introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system. Let us try to be clear. There is a fundamental difference between the system we are about to discuss here and any other talking machine (as a cassette-player for example) in the sense that we are interested in the automatic production of new sentences. This definition still needs some refinements. Systems that simply concatenate isolated words or parts of sentences, denoted as Voice Response Systems, are only applicable when a limited vocabulary is required (typically a few one hundreds of words), and when the sentences to be pronounced respect a very restricted structure, as is the case for the announcement of arrivals in train stations for instance. In the context of TTS synthesis, it is impossible (and luckily useless) to record and store all the words of the language. It is thus more suitable to define Text-To-Speech as the automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter.

At first sight, this task does not look too hard to perform. After all, is not the human being potentially able to correctly pronounce an unknown sentence, even from his childhood? We all have, mainly unconsciously, a deep knowledge of the reading rules of our mother tongue. They were transmitted to us, in a simplified form, at primary school, and we improved them year after year. However, it would be a bold claim indeed to say that it is only a short step before the computer is likely to equal the human being in that respect. Despite the present state of our knowledge and techniques and the progress recently accomplished in the fields of Signal Processing and Artificial Intelligence, we would have to express some reservations. As a matter of fact, the reading process draws from the furthest depths, often unthought-of, of the human intelligence.

4.2 Applications

Each and every synthesizer is the result of a particular and original imitation of the human reading capability, submitted to technological and imaginative constraints that are characteristic of the time of its creation. The concept of high quality TTS synthesis appeared in the mid eighties, as a result of important developments in speech synthesis and natural language processing techniques, mostly due to the emergence of new technologies (Digital Signal and Logical Inference Processors). It is now a must for the speech products family expansion. Potential applications of High Quality TTS Systems are indeed numerous. Here are some examples:

- **Telecommunications services.** TTS systems make it possible to access textual information over the telephone. Knowing that about 70 % of the telephone calls actually require very little interactivity, such a prospect is worth being considered. Texts might range from simple messages, such as local cultural events not to miss (cinemas, theatres), to huge databases which can hardly be read and stored as digitized speech. Queries to such information retrieval systems could be put through the user's voice (with the help of a speech recognizer), or through the telephone keyboard (with DTMF systems). One could even imagine that our (artificially) intelligent machines could speed up the query when needed, by providing lists of keywords, or even summaries. In this connection, AT&T has recently organized a series of consumer tests for some promising telephone services. They include: Who's Calling (get the spoken name of your caller before being connected and hang up to avoid the call), Integrated Messaging (have your electronic mail or facsimiles being automatically read over the telephone), Telephone Relay Service (have a telephone conversation with speech or hearing impaired persons thanks to ad hoc text-to-voice and voice-to-text conversion), and Automated Caller Name and Address (a computerized version of the "reverse directory"). These applications have proved acceptable, and even popular, provided the intelligibility of the synthetic utterances was high enough. Naturalness was not a major issue in most cases.
- **Language education.** High Quality TTS synthesis can be coupled with a Computer Aided Learning system, and provide a helpful tool to learn a new language. To

our knowledge, this has not been done yet, given the relatively poor quality available with commercial systems, as opposed to the critical requirements of such tasks.

- **Aid to handicapped persons.** Voice handicaps originate in mental or motor/sensation disorders. Machines can be an invaluable support in the latter case: with the help of an especially designed keyboard and a fast sentence assembling program, synthetic speech can be produced in a few seconds to remedy these impediments. Astro-physician Stephen Hawking gives all his lectures in this way. The aforementioned Telephone Relay Service is another example. Blind people also widely benefit from TTS systems, when coupled with Optical Recognition Systems (OCR), which give them access to written information. The market for speech synthesis for blind users of personal computers will soon be invaded by mass-market synthesizers bundled with sound cards.
- **Talking books and toys.** The toy market has already been touched by speech synthesis. Many speaking toys have appeared, under the impulse of the innovative 'Magic Spell' from Texas Instruments. The poor quality available inevitably restrains the educational ambition of such products. High Quality synthesis at affordable prices might well change this.
- **Vocal Monitoring.** In some cases, oral information is more efficient than written messages. The appeal is stronger, while the attention may still focus on other visual sources of information. Hence the idea of incorporating speech synthesizers in measurement or control systems.
- **Multimedia, man-machine communication.** In the long run, the development of high quality TTS systems is a necessary step (as is the enhancement of speech recognizers) towards more complete means of communication between men and computers. Multimedia is a first but promising move in this direction.
- **Fundamental and applied research.** TTS synthesizers possess a very peculiar feature which makes them wonderful laboratory tools for linguists: they are completely under control, so that repeated experiences provide identical results (as is hardly the case with human beings). Consequently, they allow to investigate the efficiency of into native and rhythmic models. A particular type of TTS systems, which are based on a description of the vocal tract through its resonant frequencies (its formants) and denoted as formant synthesizers, has also been extensively used by phoneticians to study speech in terms of

acoustical rules. In this manner, for instance, articulatory constraints have been enlightened and formally described.

4.3 Working of TTS system

From now on, it should be clear that a reading machine would hardly adopt a processing scheme as the one naturally taken up by humans, whether it was for language analysis or for speech production itself. Vocal sounds are inherently governed by the partial differential equations of fluid mechanics, applied in a dynamic case since our lung pressure, glottis tension, and vocal and nasal tracts configuration evolve with time. These are controlled by our cortex, which takes advantage of the power of its parallel structure to extract the essence of the text read: its meaning. Even though, in the current state of the engineering art, building a Text-To-Speech synthesizer on such intricate models is almost scientifically conceivable (intensive research on articulatory synthesis, neural networks, and semantic analysis give evidence of it), it would result anyway in a machine with a very high degree of (possibly avoidable) complexity, which is not always compatible with economical criteria. After all, flies do not flap their wings!

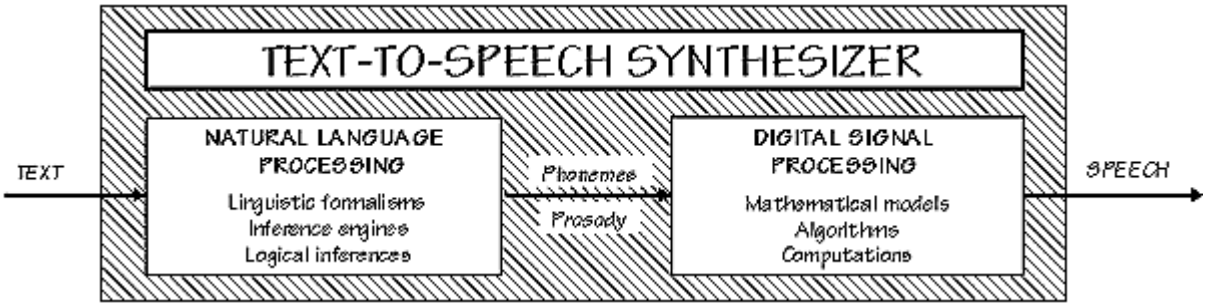


Fig 4.1 General function diagram of a TTS system

Figure 4.1 introduces the functional diagram of a very general TTS synthesizer. As for human reading, it comprises a Natural Language Processing module (NLP), capable of producing a phonetic transcription of the text read, together with the desired intonation and rhythm (often termed as prosody), and a Digital Signal Processing module (DSP), which transforms the symbolic information it receives into speech. But the formalisms and algorithms applied often manage, thanks to a judicious use of mathematical and

linguistic knowledge of developers, to short-circuit certain processing steps. This is occasionally achieved at the expense of some restrictions on the text to pronounce, or results in some reduction of the "emotional dynamics" of the synthetic voice (at least in comparison with human performances), but it generally allows to solve the problem in real time with limited memory requirements.

4.4 The NLP component

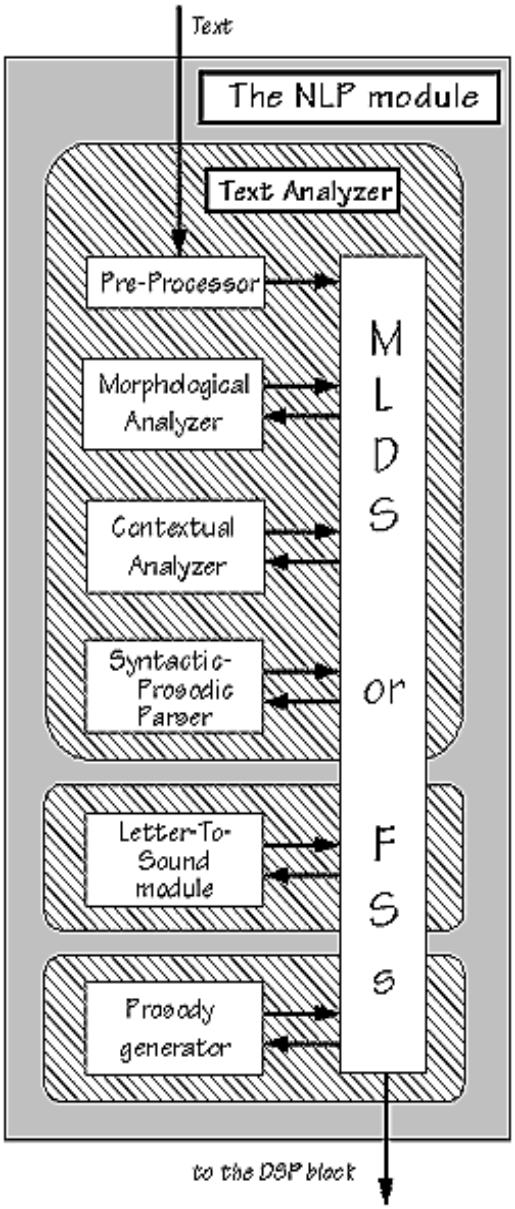


Fig 4.2: The NLP module of a general Text to Speech conversion system.

Figure 4.2 introduces the skeleton of a general NLP module for TTS purposes. One immediately notices that, in addition with the expected letter-to-sound and prosody generation blocks, it comprises a morpho-syntactic analyzer, underlying the need for some syntactic processing in a high quality Text-To-Speech system. Indeed, being able to reduce a given sentence into something like the sequence of its parts-of-speech, and to further describe it in the form of a syntax tree, which unveils its internal structure, is required for at least two reasons:

A) Accurate phonetic transcription can only be achieved provided the part of speech category of some words is available, as well as if the dependency relationship between successive words is known.

B) Natural prosody heavily relies on syntax. It also obviously has a lot to do with semantics and pragmatics, but since very few data is currently available on the generative aspects of this dependence, TTS systems merely concentrate on syntax. Yet few of them are actually provided with full disambiguation and structuration capabilities.

4.4.1 Text analysis

The text analysis block is itself composed of:

- **A pre-processing module**, which organizes the input sentences into manageable lists of words. It identifies numbers, abbreviations, acronyms and idiomatic and transforms them into full text when needed. An important problem is encountered as soon as the character level: that of punctuation ambiguity (including the critical case of sentence end detection). It can be solved, to some extent, with elementary regular grammars.
- **A morphological analysis module**, the task of which is to propose all possible part of speech categories for each word taken individually, on the basis of their spelling. Inflected, derived, and compound words are decomposed into their elementary graphemic units (their morphs) by simple regular grammars exploiting lexicons of stems and affixes.
- **The contextual analysis module** considers words in their context, which allows it to reduce the list of their possible part of speech categories to a very restricted number

of highly probable hypotheses, given the corresponding possible parts of speech of neighboring words. This can be achieved either with n-grams, which describe local syntactic dependences in the form of probabilistic finite state automata (i.e. as a Markov model), to a lesser extent with multi-layer perceptrons (i.e., neural networks) trained to uncover contextual rewrite rules, or with local, non-stochastic grammars provided by expert linguists or automatically inferred from a training data set with classification and regression tree.

- **Syntactic-prosodic parser**, which examines the remaining search space and finds the text structure (i.e. its organization into clause and phrase-like constituents) which more closely relates to its expected prosodic realization.

4.4.2 Automatic phonetization

The Letter-To-Sound (LTS) module is responsible for the automatic determination of the phonetic transcription of the incoming text. It thus seems, at first sight, that its task is as simple as performing the equivalent of a dictionary look-up! From a deeper examination, however, one quickly realizes that most words appear in genuine speech with several phonetic transcriptions, many of which are not even mentioned in pronunciation dictionaries. Namely:

A) Pronunciation dictionaries refer to word roots only. They do not explicitly account for morphological variations (i.e. plural, feminine, conjugations, especially for highly inflected languages, such as French), which therefore have to be dealt with by a specific component of phonology, called morphophonology.

B) Some words actually correspond to several entries in the dictionary, or more generally to several morphological analyses, generally with different pronunciations. This is typically the case of heterophonic homographs, i.e. words that are pronounced differently even though they have the same spelling, as for 'record', constitute by far the most tedious class of pronunciation ambiguities. Their correct pronunciation generally depends on their part-of-speech and most frequently contrasts verbs and non-verbs , as for 'contrast' (verb/noun) or 'intimate' (verb/adjective), although it may also be based on syntactic features, as for 'read' (present/past)

C) Pronunciation dictionaries merely provide something that is closer to a phonemic transcription than from a phonetic one (i.e. they refer to phonemes rather than to phones). As denoted by Withgott and Chen : "while it is relatively straightforward to build computational models for morphophonological phenomena, such as producing the dictionary pronunciation of 'electricity' given a base form 'electric', it is another matter to model how that pronunciation actually sounds". Consonants, for example, may reduce or delete in clusters, a phenomenon termed as consonant cluster simplification, as in 'softness' in which [t] fuses in a single gesture with the following [n].

D) Words embedded into sentences are not pronounced as if they were isolated. Surprisingly enough, the difference does not only originate in variations at word boundaries (as with phonetic liaisons), but also on alternations based on the organization of the sentence into non-lexical units, that is whether into groups of words (as for phonetic lengthening) or into non-lexical parts thereof (many phonological processes, for instance, are sensitive to syllable structure).

E) Finally, not all words can be found in a phonetic dictionary: the pronunciation of new words and of many proper names has to be deduced from the one of already known words.

Clearly, points A and B heavily rely on a preliminary morphosyntactic (and possibly semantic) analysis of the sentences to read. To a lesser extent, it also happens to be the case for point C as well, since reduction processes are not only a matter of context-sensitive phonation, but they also rely on morphological structure and on word grouping, that is on morphosyntax. Point D puts a strong demand on sentence analysis, whether syntactic or metrical, and point E can be partially solved by addressing morphology and/or by finding graphemic analogies between words.

It is then possible to organize the task of the LTS module in many ways (Fig.4.3), often roughly classified into dictionary-based and rule-based strategies, although many intermediate solutions exist.

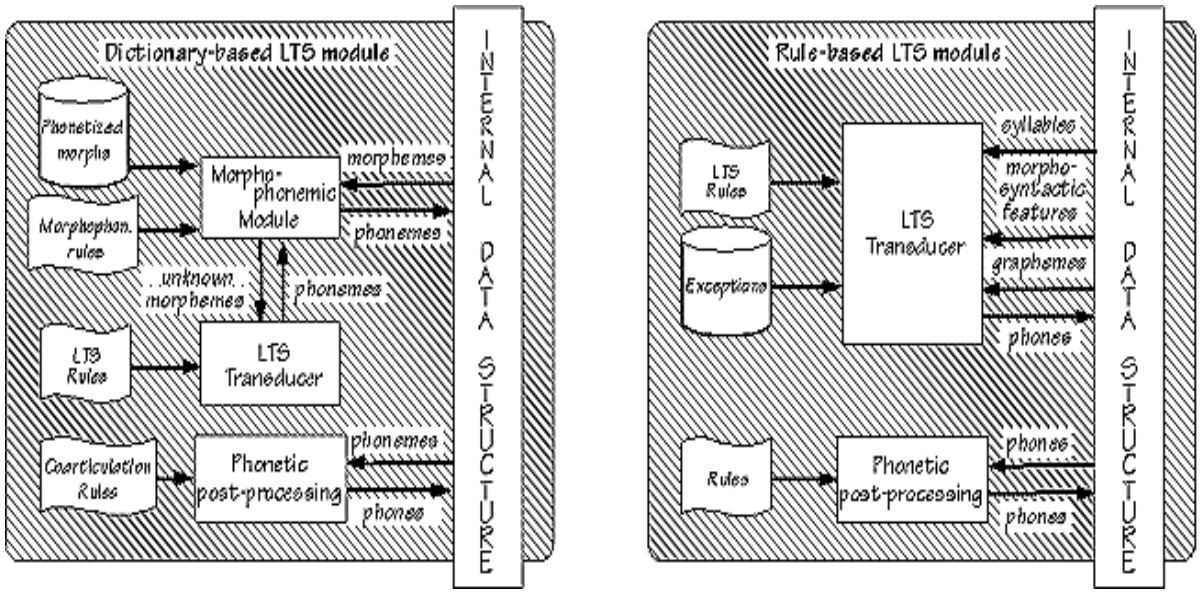


Fig 4.3: Dictionary-based (left) versus rule-based (right) phonetization.

Dictionary-based solutions consist of storing a maximum of phonological knowledge into a lexicon. In order to keep its size reasonably small, entries are generally restricted to morphemes, and the pronunciation of surface forms is accounted for by inflectional, derivational, and compounding morphophonemic rules which describe how the phonetic transcriptions of their morphemic constituents are modified when they are combined into words. Morphemes that cannot be found in the lexicon are transcribed by rule. After a first phonemic transcription of each word has been obtained, some phonetic post-processing is generally applied, so as to account for co-articulatory smoothing phenomena. This approach has been followed by the MITTALK system from its very first day. A dictionary of up to 12,000 morphemes covered about 95% of the input words. The AT&T Bell Laboratories TTS system follows the same guideline, with an augmented morpheme lexicon of 43,000 morphemes.

A rather different strategy is adopted in rule-based transcription systems, which transfer most of the phonological competence of dictionaries into a set of letter-to-sound (or grapheme-to-phoneme) rules. This time, only those words that are pronounced in such a particular way that they constitute a rule on their own are stored in an exceptions dictionary. Notice that, since many exceptions are found in the most frequent words, a

reasonably small exceptions dictionary can account for a large fraction of the words in a running text. In English, for instance, 2000 words typically suffice to cover 70% of the words in text.

It has been argued in the early days of powerful dictionary-based methods that they were inherently capable of achieving higher accuracy than letter-to-sound rules [Coker et al 90], given the availability of very large phonetic dictionaries on computers. On the other hand, considerable efforts have recently been made towards designing sets of rules with a very wide coverage (starting from computerized dictionaries and adding rules and exceptions until all words are covered. Clearly, some trade-off is inescapable. Besides, the compromise is language-dependent, given the obvious differences in the reliability of letter-to-sound correspondences for different languages.

4.4.3 Prosody generation

The term prosody refers to certain properties of the speech signal which are related to audible changes in pitch, loudness, and syllable length. Prosodic features have specific functions in speech communication (see Fig. 4.4). The most apparent effect of prosody is that of focus. For instance, there are certain pitch events which make a syllable stand out within the utterance, and indirectly the word or syntactic group it belongs to will be highlighted as an important or new component in the meaning of that utterance. The presence of a focus marking may have various effects, such as contrast, depending on the place where it occurs, or the semantic context of the utterance.

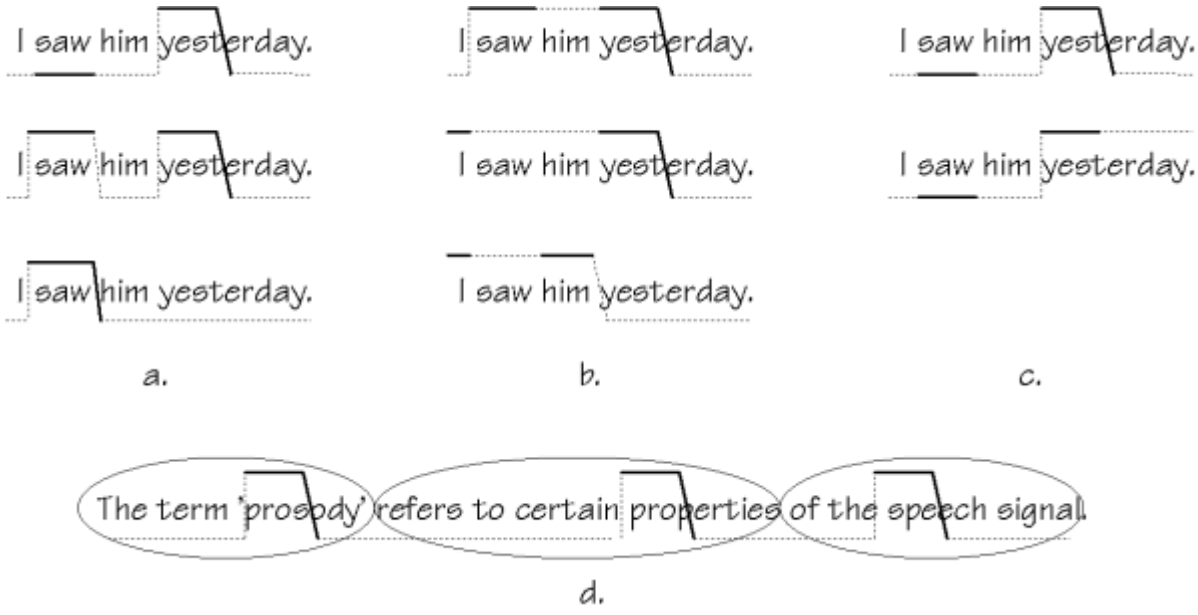


Fig 4.4: Different kinds of information provided by intonation (lines indicate pitch movements; solid lines indicate stress).

a. Focus or given/new information;

b. Relationships between words (saw-yesterday; I-yesterday; I-him)

c. Finality (top) or continuation (bottom), as it appears on the last syllable;

d. Segmentation of the sentence into groups of syllables.

Although maybe less obvious, there are other, more systematic or general functions. Prosodic features create a segmentation of the speech chain into groups of syllables, or, put the other way round, they give rise to the grouping of syllables and words into larger chunks. Moreover, there are prosodic features which indicate relationships between such groups, indicating that two or more groups of syllables are linked in some way. This grouping effect is hierarchical, although not necessarily identical to the syntactic structuring of the utterance.

The key idea is that the "correct" syntactic structure, the one that precisely requires some semantic and pragmatic insight, is not essential for producing such a prosody.

With these considerations in mind, it is not surprising that commercially developed TTS system have emphasized coverage rather than linguistic sophistication, by concentrating

their efforts on text analysis strategies aimed to segment the surface structure of incoming sentences, as opposed to their syntactically, semantically, and pragmatically related deep structure. The resulting syntactic-prosodic descriptions organize sentences in terms of prosodic groups strongly related to phrases (and therefore also termed as minor or intermediate phrases), but with a very limited amount of embedding, typically a single level for these minor phrases as parts of higher-order prosodic phrases (also termed as major or intonational phrases, which can be seen as a prosodic-syntactic equivalent for clauses) and a second one for these major phrases as parts of sentences, to the extent that the related major phrase boundaries can be safely obtained from relatively simple text analysis methods. In other words, they focus on obtaining an acceptable segmentation and translate it into the continuation or finality marks of Fig. 4.c, but ignore the relationships or contrastive meaning of Fig. 4.4.a and b.

A (minor) prosodic phrase = a sequence of chinks followed by a sequence of chunks

in which chinks and chunks belong to sets of words which basically correspond to function and content words, respectively, with the difference that objective pronouns (like 'him' or 'them') are seen as chunks and that tensed verb forms (such as 'produced') are considered as chinks. They show that this approach produces efficient grouping in most cases, slightly better actually than the simpler decomposition into sequences of function and content words, as shown in the example below:

function words / content words	chinks / chunks
I asked	I asked them
them if they were going home	if they were going home
to Patiala	to Patiala
and they said yes	and they said yes
and anticipated	and anticipated one more stop
one more stop	before getting home
before getting home	

Other, more sophisticated approaches include syntax-based expert systems and automatic, corpus-based methods as with the classification and regression tree (CART) techniques of Hirschberg .

Once the syntactic-prosodic structure of a sentence has been derived, it is used to obtain the precise duration of each phoneme (and of silences), as well as the intonation to apply on them. This last step, however, is not straightforward either. It requires to formalize a lot of phonetic or phonological knowledge, either obtained from experts or automatically acquired from data with statistical methods.

4.5 The DSP component

Intuitively, the operations involved in the DSP module are the computer analogue of dynamically controlling the articulatory muscles and the vibratory frequency of the vocal folds so that the output signal matches the input requirements. In order to do it properly, the DSP module should obviously, in some way, take articulatory constraints into account, since it has been known for a long time that phonetic transitions are more important than stable states for the understanding of speech. This, in turn, can be basically achieved in two ways:

- Explicitly, in the form of a series of rules which formally describe the influence of phonemes on one another;
- Implicitly, by storing examples of phonetic transitions and co-articulations into a speech segment database, and using them just as they are, as ultimate acoustic units (i.e. in place of phonemes).

Two main classes of TTS systems have emerged from this alternative, which quickly turned into synthesis philosophies given the divergences they present in their means and objectives: synthesis-by-rule and synthesis-by-concatenation.

4.5.1 Rule-based synthesizers:

Rule-based synthesizers are mostly in favor with phoneticians and phonologists, as they constitute a cognitive, generative approach of the phonation mechanism. The broad spreading of the Klatt synthesizer, for instance, is principally due to its invaluable assistance in the study of the characteristics of natural speech, by analytic listening of rule-synthesized speech. What is more, the existence of relationships between articulatory parameters and the inputs of the Klatt model make it a practical tool for investigating physiological constraints.

For historical and practical reasons (mainly the need for a physical interpretability of the model), rule synthesizers always appear in the form of formant synthesizers. These describe speech as the dynamic evolution of up to 60 parameters [Stevens 90], mostly related to formant and anti-formant frequencies and bandwidths together with glottal waveforms. Clearly, the large number of (coupled) parameters complicates the analysis stage and tends to produce analysis errors. What is more, formant frequencies and bandwidths are inherently difficult to estimate from speech data. The need for intensive trials and errors in order to cope with analysis errors, makes them time-consuming systems to develop (several years are commonplace). Yet, the synthesis quality achieved up to now reveals typical buzzyness problems, which originate from the rules themselves: introducing a high degree of naturalness is theoretically possible, but the rules to do so are still to be discovered.

Rule-based synthesizers remain, however, a potentially powerful approach to speech synthesis. They allow, for instance, to study speaker-dependent voice features so that switching from one synthetic voice into another can be achieved with the help of specialized rules in the rule database. Following the same idea, synthesis-by-rule seems to be a natural way of handling the articulatory aspects of changes in speaking styles (as opposed to their prosodic counterpart, which can be accounted for by concatenation-based synthesizers as well).

4.5.2 Concatenative synthesizers

As opposed to rule-based ones, concatenative synthesizers possess a very limited knowledge of the data they handle: most of it is embedded in the segments to be chained up. This clearly appears in figure 4.5, where all the operations that could indifferently be used in the context of a music synthesizer (i.e. without any explicit reference to the inner nature of the sounds to be processed) have been grouped into a sound processing block, as opposed to the upper speech processing block whose design requires at least some understanding of phonetics.

- **Database operation:** A series of preliminary stages have to be fulfilled before the synthesizer can produce its first utterance. At first, segments are chosen so as to minimize future concatenation problems. A combination of diphones (i.e. units that begin in the middle of the stable state of a phone and end in the middle of the following one), half-syllables, and triphones (which differ from diphones in that they include a complete central phone) are often chosen as speech units, since they involve most of the transitions and co-articulations while requiring an affordable amount of memory. When a complete list of segments has emerged, a corresponding list of words is carefully completed, in such a way that each segment appears at least once (twice is better, for security). Unfavorable positions like inside stressed syllables or in strongly reduced (i.e. over-co-articulated) contexts, are excluded. A corpus is then digitally recorded and stored, and the elected segments are spotted, either manually with the help of signal visualization tools, or automatically thanks to segmentation algorithms, the decisions of which are checked and corrected interactively. A segment database finally centralizes the results, in the form of the segment names, waveforms, durations, and internal sub-splitting. In the case of diphones, for example, the position of the border between phones should be stored, so as to be able to modify the duration of one half-phone without affecting the length of the other one.

Segments are then often given a parametric form, in the form of a temporal sequence of vectors of parameters collected at the output of a speech analyzer and stored in a

parametric segment database. The advantage of using a speech model originates in the fact that:

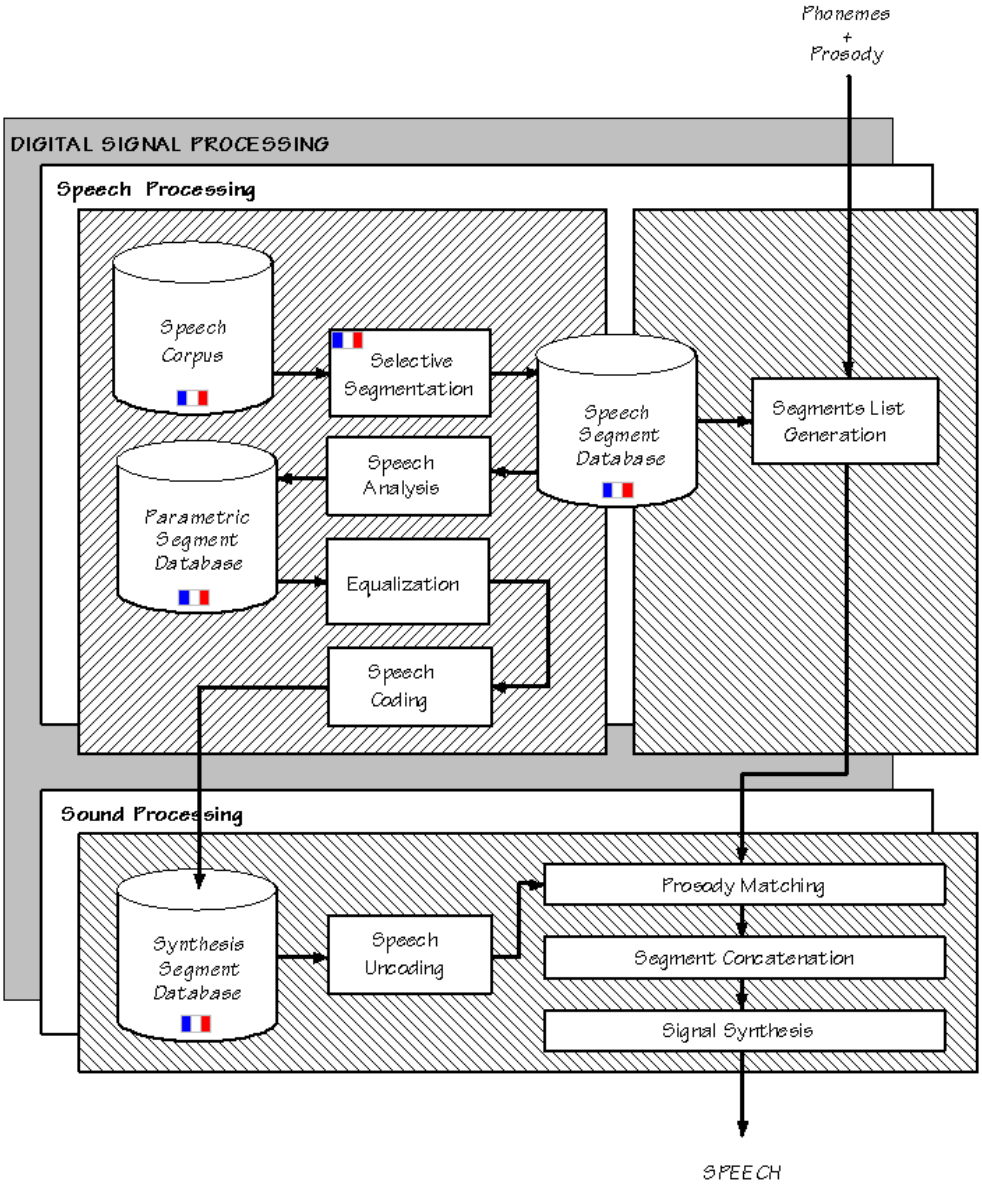


Figure 4.5: A general concatenation-based synthesizer.

The upper left hatched block corresponds to the development of the synthesizer (i.e. it is processed once for all). Other blocks correspond to run-time operations. Language-dependent operations and data are indicated by a flag.

- Well chosen speech models allow data size reduction, an advantage which is hardly negligible in the context of concatenation-based synthesis given the amount of data to be stored. Consequently, the analyzer is often followed by a parametric speech coder.

- A number of models explicitly separate the contributions of respectively the source and the vocal tract, an operation which remains helpful for the pre-synthesis operations: prosody matching and segments concatenation.

Indeed, the actual task of the synthesizer is to produce, in real-time, an adequate sequence of concatenated segments, extracted from its parametric segment database and the prosody of which has been adjusted from their stored value, i.e. the intonation and the duration they appeared with in the original speech corpus, to the one imposed by the language processing module. Consequently, the respective parts played by the prosody matching and segments concatenation modules are considerably alleviated when input segments are presented in a form that allows easy modification of their pitch, duration, and spectral envelope, as is hardly the case with crude waveform samples.

Since segments to be chained up have generally been extracted from different words, that is in different phonetic contexts, they often present amplitude and timbre mismatches. Even in the case of stationary vocalic sounds, for instance, a rough sequencing of parameters typically leads to audible discontinuities. These can be coped with during the constitution of the synthesis segments database, thanks to an equalization in which related endings of segments are imposed similar amplitude spectra, the difference being distributed on their neighborhood. In practice, however, this operation, is restricted to amplitude parameters: the equalization stage smoothly modifies the energy levels at the beginning and at the end of segments, in such a way as to eliminate amplitude mismatches (by setting the energy of all the phones of a given phoneme to their average value). In contrast, timbre conflicts are better tackled at run-time, by smoothing individual couples of segments when necessary rather than equalizing them once for all, so that some of the phonetic variability naturally introduced by co-articulation is still maintained. In practice, amplitude equalization can be performed either before or after

speech analysis (i.e. on crude samples or on speech parameters).Once the parametric segment database has been completed, synthesis itself can begin.

- **Speech Synthesis:** A sequence of segments is first deduced from the phonemic input of the synthesizer, in a block termed as segment list generation, which interfaces the NLP and DSP modules. Once prosodic events have been correctly assigned to individual segments, the prosody matching module queries the synthesis segment database for the actual parameters, adequately uncoded, of the elementary sounds to be used, and adapts them one by one to the required prosody. The segment concatenation block is then in charge of dynamically matching segments to one another, by smoothing discontinuities. Here again, an adequate modelization of speech is highly profitable, provided simple interpolation schemes performed on its parameters approximately correspond to smooth acoustical transitions between sounds. The resulting stream of parameters is finally presented at the input of a synthesis block, the exact counterpart of the analysis one. Its task is to produce speech.

- **Segmental Quality:** The efficiency of concatenative synthesizers to produce high quality speech is mainly subordinated to the type of segments chosen.

1. Segments should obviously exhibit some basic properties:
    - They should allow to account for as many co-articulatory effects as possible.
    - Given the restricted smoothing capabilities of the concatenation block, they should be easily connectable.
    - Their number and length should be kept as small as possible.
    - On the other hand, longer units decrease the density of concatenation points, therefore providing better speech quality. Similarly, an obvious way of accounting for articulatory phenomena is to provide many variants for each phoneme. This is clearly in contradiction with the limited memory constraint. Some trade-off is necessary. Diphones are often chosen. They are not too numerous (about 1200 for French, including lots of phoneme sequences that are only encountered at word boundaries, for 3 minutes of speech, i.e. approximately 5 Mbytes of 16 bits samples at 16 kHz) and they do

incorporate most phonetic transitions. No wonder then that they have been extensively used. They imply, however, a high density of concatenation points (one per phoneme), which reinforces the importance of an efficient concatenation algorithm. Besides, they can only partially account for the many co-articulatory effects of a spoken language, since these often affect a whole phone rather than just its right or left halves independently. Such effects are especially patent when somewhat transient phones, such as liquids and (worst of all) semi-vowels, are to be connected to each other. Hence the use of some larger units as well, such as triphones.

2. The model of speech signal, to which the analysis and synthesis algorithms refer.

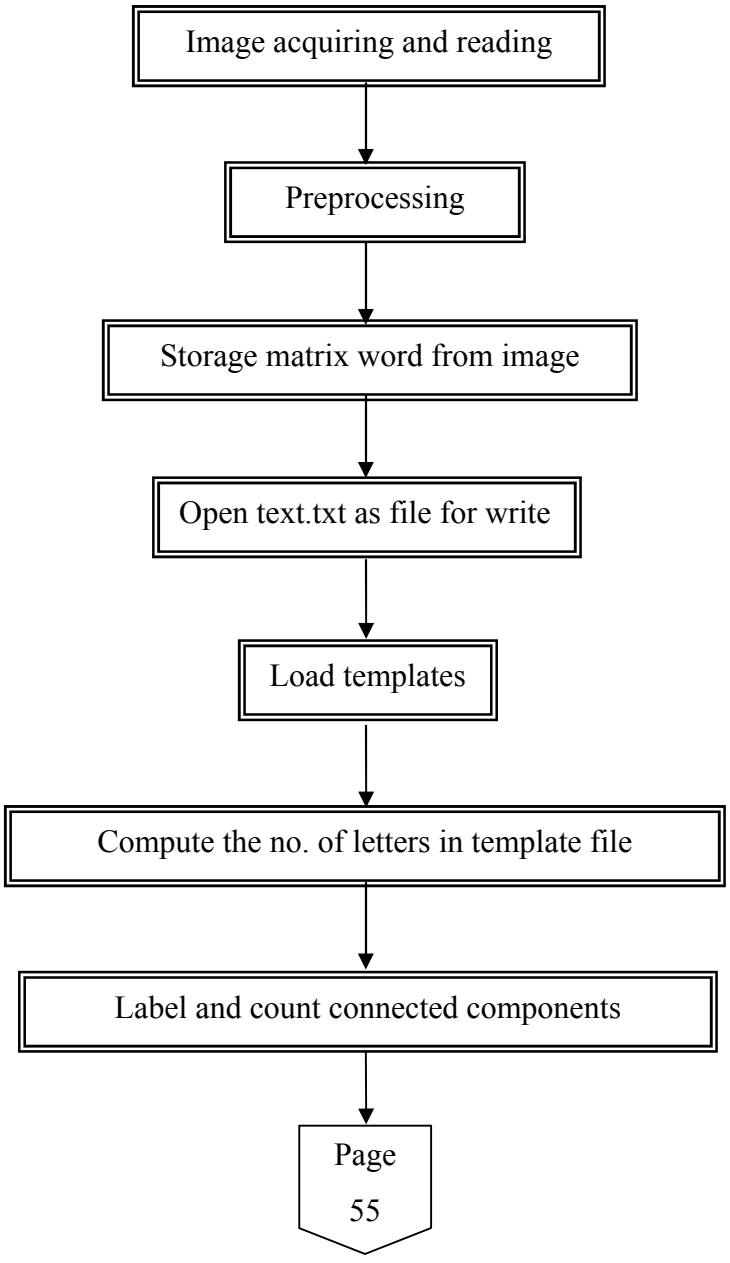
The models used in the context of concatenative synthesis can be roughly classified into two groups, depending on their relationship with the actual phonation process. Production models provide mathematical substitutes for the part respectively played by vocal folds, nasal and vocal tracts, and by the lips radiation. Their most representative members are Linear Prediction Coding (LPC) synthesizers. On the contrary, phenomenological models intentionally discard any reference to the human production mechanism. Among these pure digital signal processing tools, spectral and time-domain approaches are increasingly encountered in TTS systems. Two leading such models exist: the hybrid Harmonic/Stochastic (H/S) model and the Time-Domain Pitch-Synchronous-OveraLap-Add (TD-PSOLA) one. The latter is a time-domain algorithm: it virtually uses no speech explicit speech model. It exhibits very interesting practical features: a very high speech quality (the best currently available) combined with a very low computational cost (7 operations per sample on the average). The hybrid Harmonic/stochastic model is intrinsically more powerful than the TD-PSOLA one, but it is also about ten times more computationally intensive. PSOLA synthesizers are now widely used in the speech synthesis community. The recently developed MBROLA algorithm even provides a time-domain algorithm which exhibits the very efficient smoothing capabilities of the H/S model (for the spectral envelope mismatches that cannot be avoided at concatenation points) as well as its very high data compression ratios (up to 10 with almost no additional computational cost) while keeping the computational complexity of PSOLA.

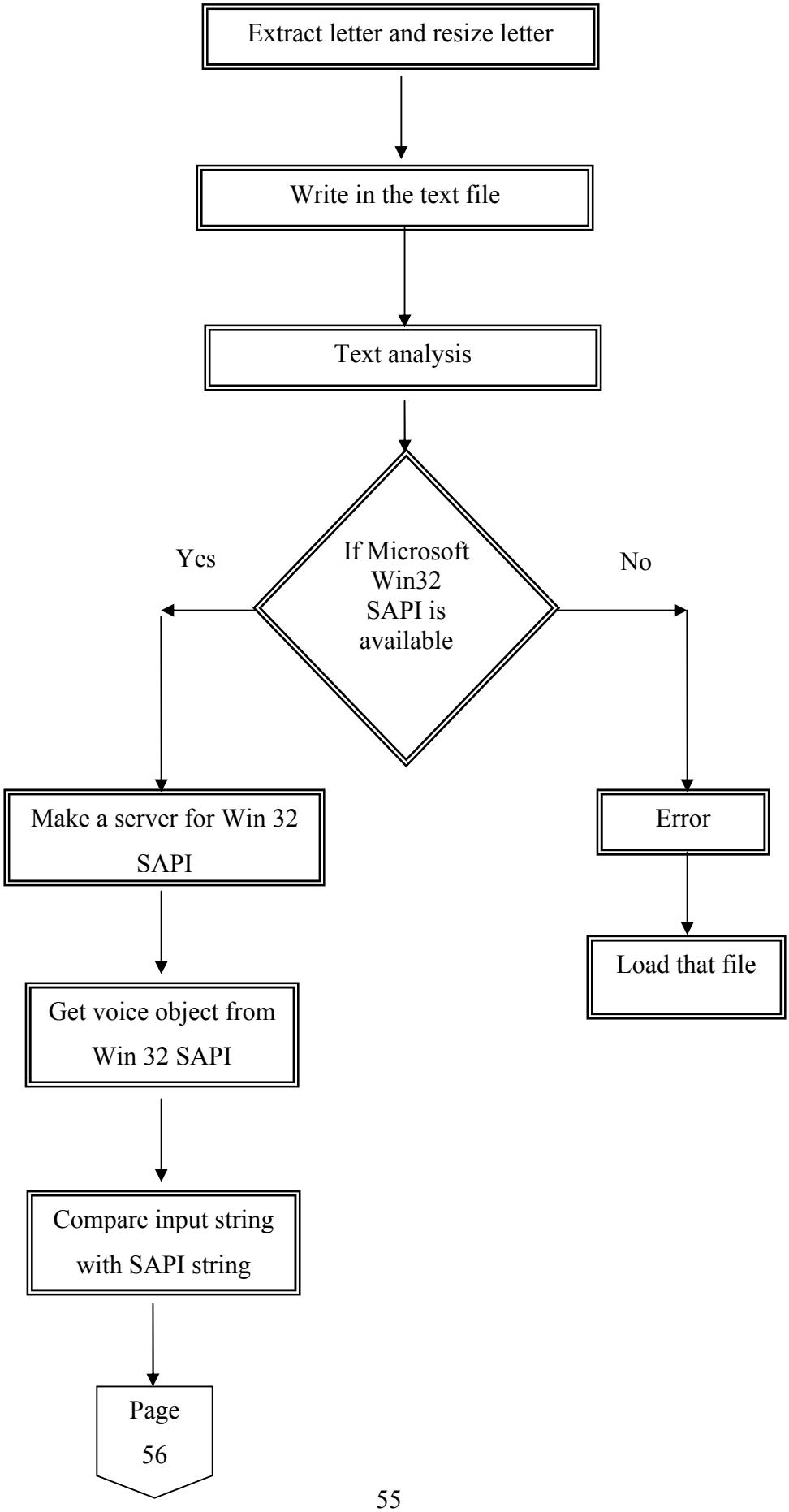
Chapter 5

Methodology

In this thesis work we convert the image into text and then text into speech. The description which has done to convert image into speech is described in this chapter. First there is a flow chart for the method and then an algorithm which is followed.

5.1 FLOW CHART




Extract letter and resize letter
Write in the text file
Text analysis
If Microsoft Win32 SAPI is available
Yes
No
Make a server for Win 32 SAPI
Error
Load that file
Get voice object from Win 32 SAPI
Compare input string with SAPI string
Page 56

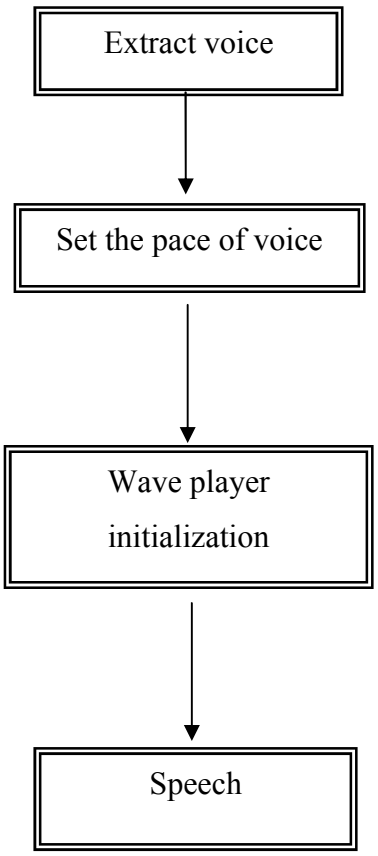


Fig 5.1: Flow chart of used methodology.

5.2 Algorithm

Step 1:- Firstly clear the screen and the image was read with the help of imread command. imread Read image from graphics file.

A = imread(filename,fmt) reads a grayscale or color image from the file specified by the string filename, where the string fmt specifies the format of the file.

Step 2:- Second step is preprocessing step. In this step firstly we convert the image into gray scale by rgb2gray command. rgb2gray Convert RGB image or colormap to grayscale.rgb2gray converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

I = rgb2gray (RGB) converts the truecolor image RGB to the grayscale intensity image I.

Then this gray image is converted into black and white image. Firstly we count the threshold in gray image then according to that threshold we convert it into black and white image. Then we remove all the objects from less than 30 pixels by bwareaopen command. bwareaopen Morphologically open binary image (remove small objects).

BW2 = bwareaopen (BW,P) removes from a binary image all connected components (objects) that have fewer than P pixels, producing another binary image BW2.

Step 3:- In this step we convert the black and white image into word matrix for further calculation.

Step 4:- In this step we open the text.txt as file for write by the fopen command.

FID = fopen (filename) opens the file filename for read access. filename is a string containing the name of the file to be opened.

Step 5:- In this step we load the templates so that we can match the letters with the templates.

Step 6:- In this step we compute the number of letters in template file by using the loop. In this loop firstly lines are separated from the text.

Step 7:- In this step we label and count connected components by bwlabel command. bwlabel Label connected components in 2-D binary image.

L = bwlabel (BW,N) returns a matrix L, of the same size as BW, containing labels for the connected components in BW. N can have a value of either 4 or 8, where 4 specifies 4-connected objects and 8 specifies 8-connected objects; if the argument is omitted, it defaults to 8.

Step 8:- In this step we extract letter and resize letter by imresize command. We resize letters according to templates size.

B = imresize (A, M, METHOD) returns an image that is M times the size of A. If M is between 0 and 1.0, B is smaller than A. If M is greater than 1.0, B is larger than A. If METHOD is omitted, imresize uses nearest neighbor interpolation.

Step 9:-In this step we write in the text file and concatenate the letters by using the word matrix and use the fprintf command. fprintf Write formatted data to file.

COUNT = fprintf (FID, FORMAT, A,...) formats the data in the real part of array A (and in any additional array arguments), under control of the specified FORMAT string, and writes it to the file associated with file identifier FID. COUNT is the number of bytes successfully written. FID is an integer file identifier obtained from FOPEN. It can also be 1 for standard output (the screen) or 2 for standard error. If FID is omitted, output goes to the screen.

Step 10:- In this step firstly we analysis the text, and check the condition that if Win 32 SAPI is available in the computer or not. If it is not available then error will be generated and we should load that Win 32 SAPI library in the computer.

Step 11:- This step will be execute if there is Win 32 SAPI file in the computer. Then in this step we make a new server for this file by actxserver command. actxserver create activex automation server.

H = actxserver (PROGID) will create a local or remote ActiveX automation server where PROGID is the program ID of the ActiveX object and H is the handle of the control's default interface.

Step 12:- In this step we get voice object from Win 32 SAPI by invoke command. Invoke, invoke method on object or interface, or display methods.

S = invoke (OBJ) returns structure array S containing a list of all methods supported by the object or interface OBJ along with the prototypes for these methods.

Equivalent syntax is S = OBJ.INVOKE

S = invoke (OBJ, METHOD) invokes the method specified in the string METHOD, and returns an output value, if any, in S.The data type of the return value is dependent upon the specific method being invoked and is determined by the specific control or server.

Step 13:- In this step we compare the input string with Win 32 SAPI string with strcmp command. strcmp compare strings. strcmp (S1,S2) returns logical 1 (true) if strings S1 and S2 are the same and logical 0 (false) otherwise.

STRCMP(S,T), when both inputs are character arrays returns logical 1 (true) if the arrays match in their entirety, and logical 0 (false) if they do not.

Step 14:- In this step we extract voice by firstly select the voice which are available in that library.

Step 15:- In this step we choose the pace of voice, by using the commands, nargin and nargout. Nargin command means number of function input arguments. Inside the body of a user-defined function, nargin returns the number of input arguments that were used to call the function.

Nargout command means number of function output arguments. Inside the body of a user-defined function, nargout returns the number of output arguments that were used to call the function.

Step 16:- In this step we initialize the wave player for convert the text into speech by convert uint8 to double precision.

Step 17:- Finally we get the speech for given image.

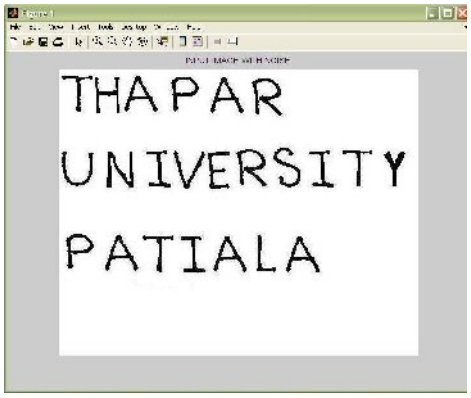
Chapter 6

Results & Discussion

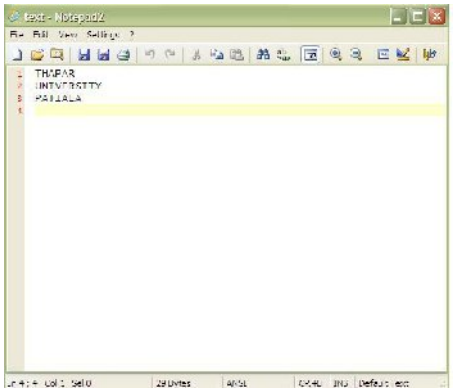
In this work a two step program is made; in first step it gives the text output according to input image with noise, then it convert that text into the speech, which is shown as periodogram.

6.1 Analysis of different images

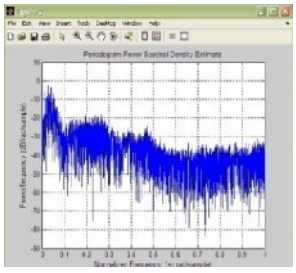
As shown in first figure, we gave the input image in which THAPAR UNIVERSITY PATIALA was written, then it is converted into text line-wise and after it the text is converted into speech, again line-wise which is shown according to text in periodogram.



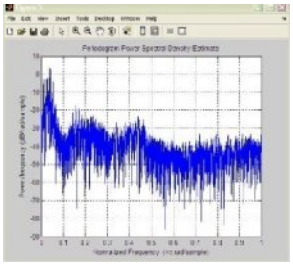
a) Input image



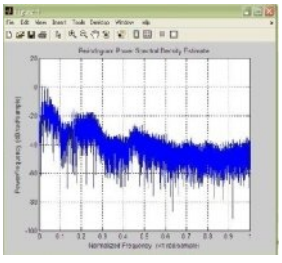
b) Output text



THAPAR



UNIVERSITY



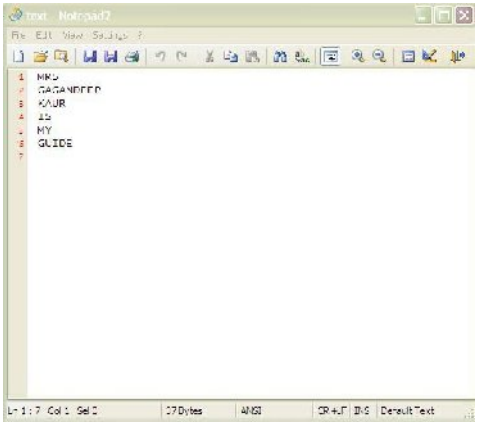
PATIALA

c) Periodogram of sound wave

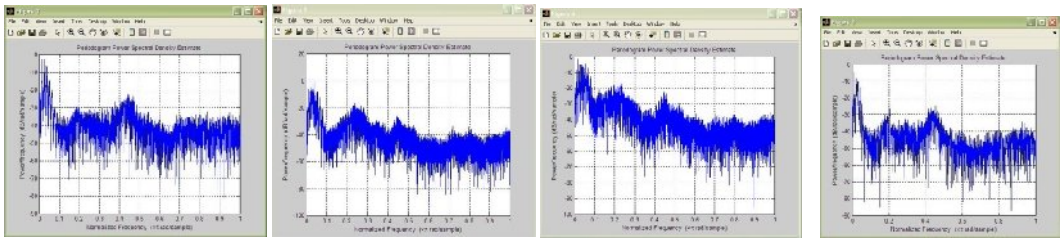
Fig 6.1 Result of image 1

In second example we increase the lines in image which are successfully converted into text and then speech which is shown in fig 6.2.

a) Input image



b) Output text



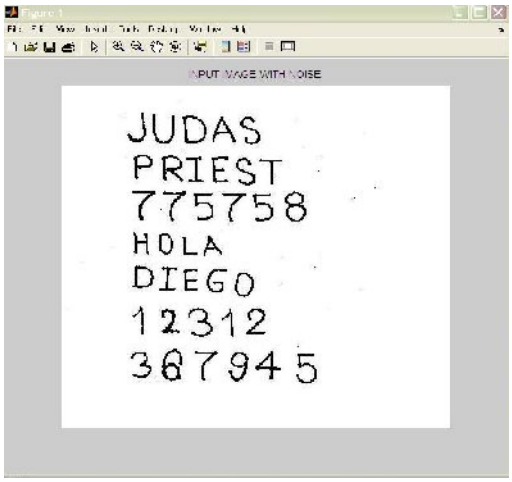
MRS GAGANDEEP KAUR IS

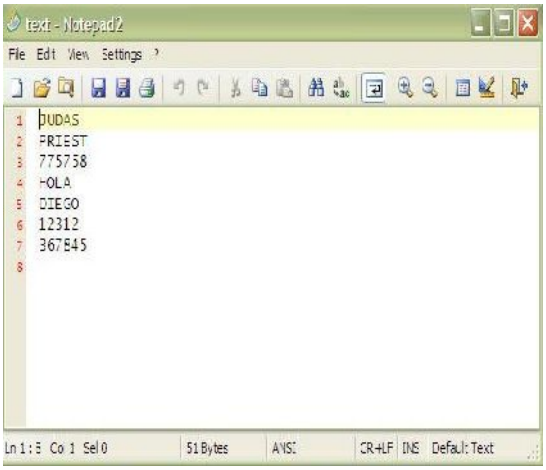
MY GUIDE

c) Periodogram of sound wave

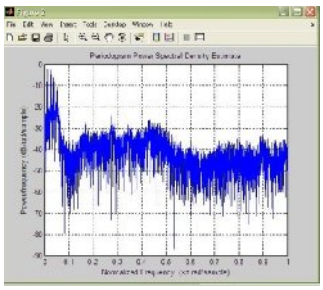
Fig 6.2 Result of image 2

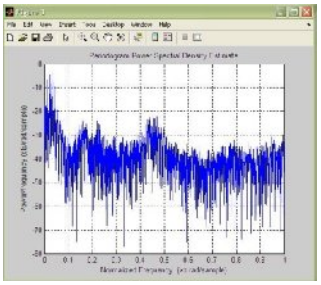
In third example we increase the line in image and insert the mathematical numbers which are successfully converted into text and then speech which is shown in fig 6.3.

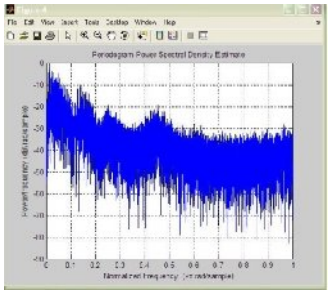




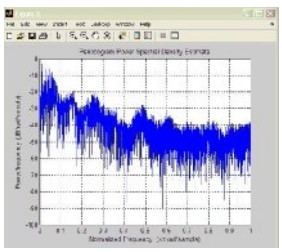
a) Input image b) Output text

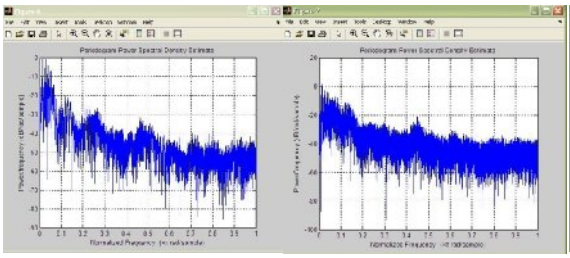


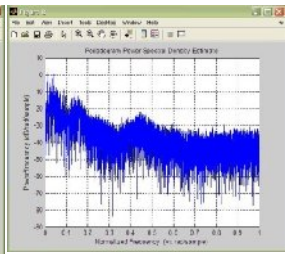




JUDAS PRIEST 77578





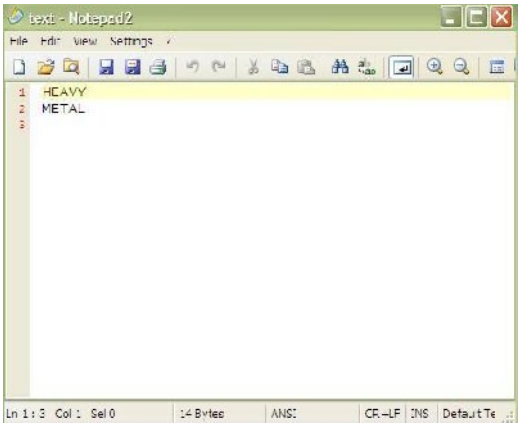


HOLA DIEGO 12312 367945

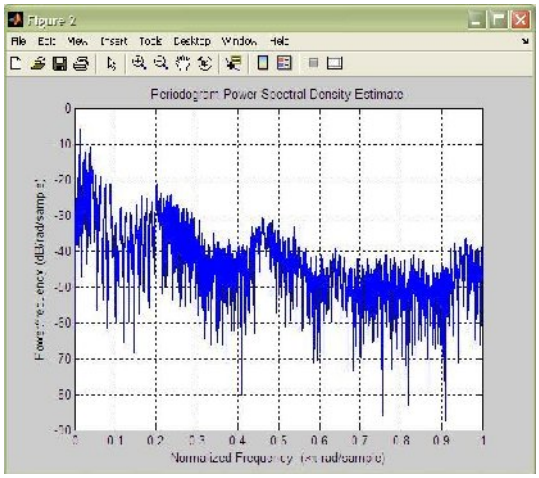
c) Periodogram of sound wave

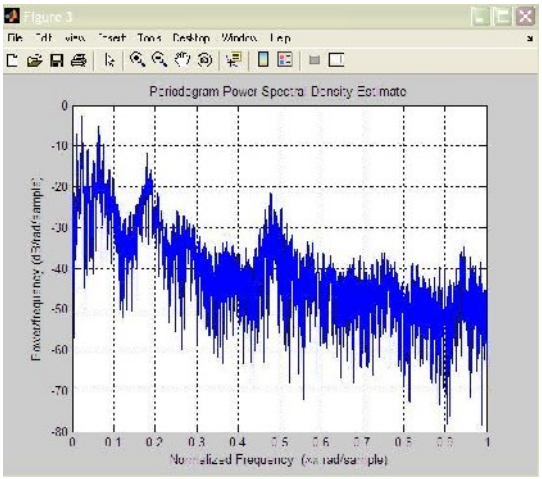
Fig 6.3 Result of image 3

In fourth example we take different type of font character image in which font of character is different than previous and again it is converted into text and then speech successfully.



a) Input image b) Output text



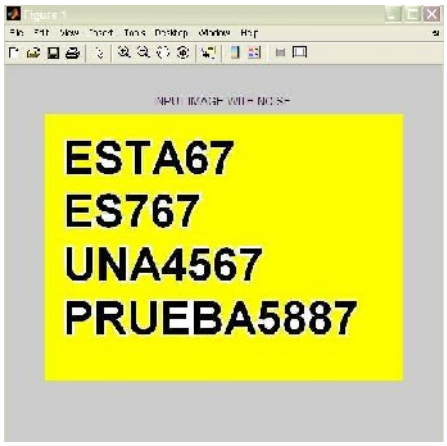


HEAVY METAL

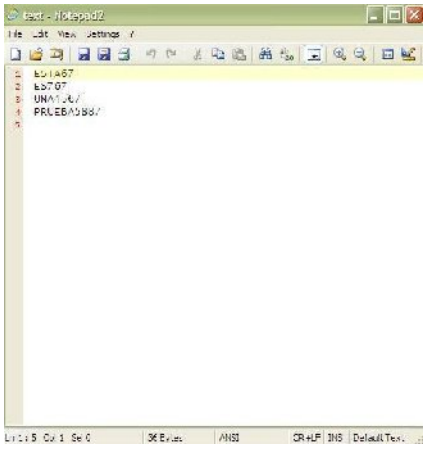
C) Periodogram of sound wave

Fig 6.4 Result of image 4

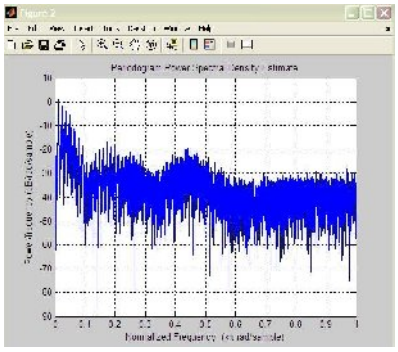
In fifth example we took a colour full image of text and it is successfully converted into text and then speech.



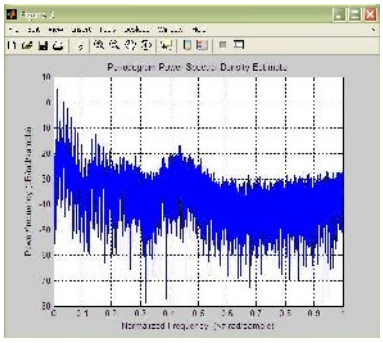
a) Input image



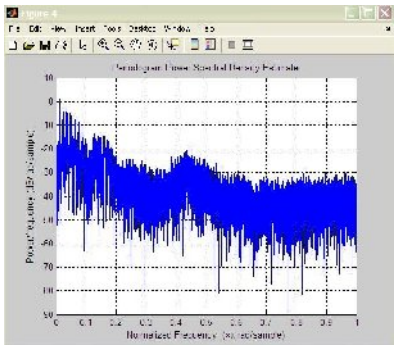
b) Output text



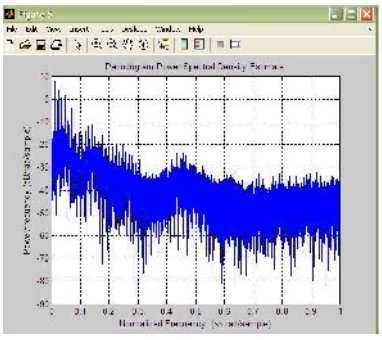
ESTA67



ES767



UNA4567

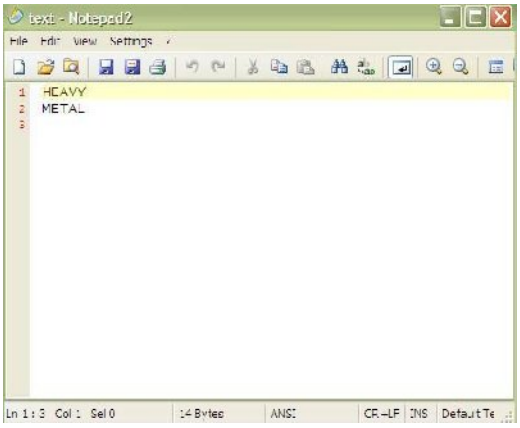


PRUEBA5887

C) Periodogram of sound wave

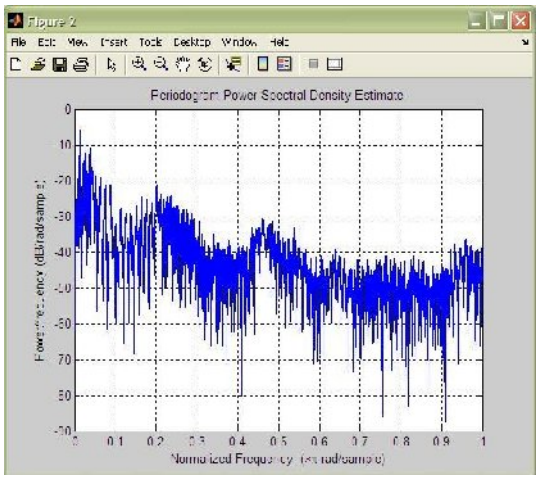
Fig 6.5 Result of image 5

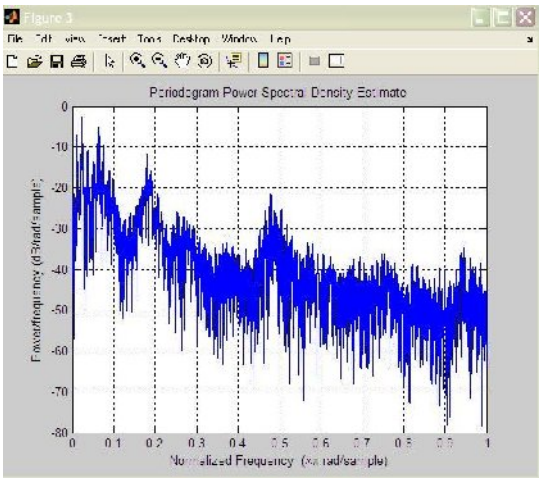
In this example we take damage character image in which image was damaged again it is converted into text and then speech successfully.



a) Input image

b) Output text





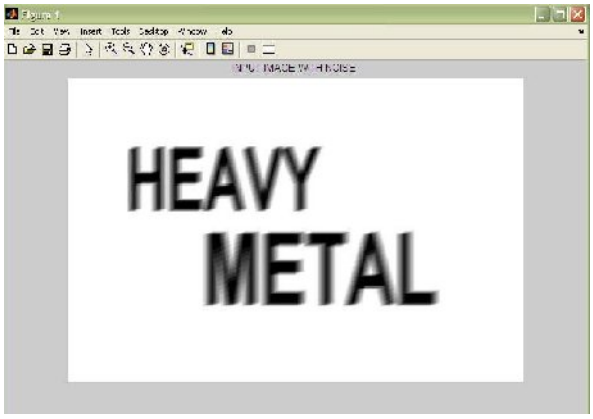
HEAVY

METAL

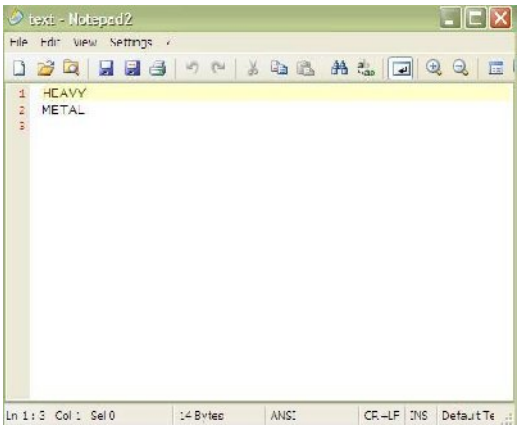
C) Periodogram of sound wave

Fig 6.6 Result of image 6

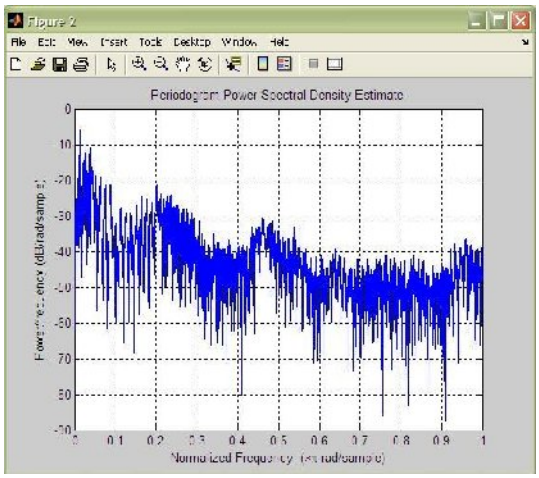
In this example we take blur character image in which the image is blur than previous one and again it is converted into text and then speech successfully.



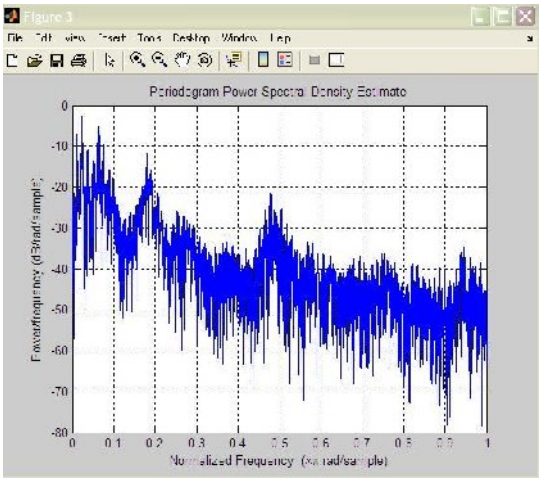
a) Input image



b) Output text



HEAVY



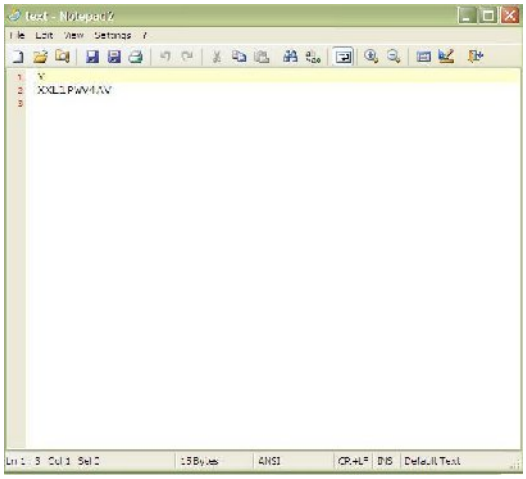
METAL

C) Periodogram of sound wave

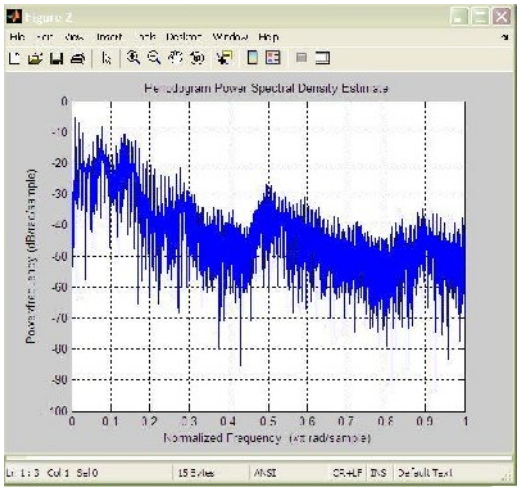
Fig 6.7 Result of image 7

In this example we take totally damaged and blur character image in which the image is blur and damaged than previous one and it is not converted into text and then speech successfully and give the wrong results.

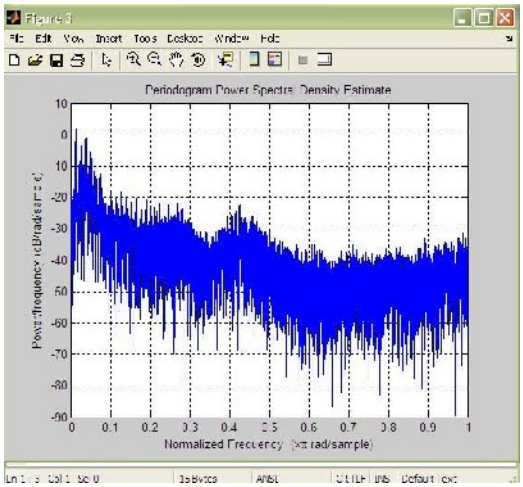
a) Input image



b) Output text



Y



XXL1PWV4AV

C) Periodogram of sound wave

Fig 6.8 Result of image 8

Chapter 7

Conclusion & Future Scope

7.1 Conclusion

Image into text and then that text into speech is converted by MATLAB. For image to text conversion firstly image is converted into gray image then black and white image and then it is converted into text by MATLAB. Microsoft Win 32 SAPI library has been used to build speech enabled applications, which retrieve the voice and audio output information available for computer. This library allows selecting the voice and audio device one would like to use. By MATLAB we can select the voices from the list and can change the pace and volume, which can be listen by installing wave player in the MATLAB. The application developed is user friendly, cost effective and applicable in the real time. The developed software has set all policies of the singles corresponding to each and every alphabet, its pronunciation methodology, the way it is used in grammar and dictionary. The pattern of the signals are processed in the system for recognition the specified frequency corresponding to particular signal is having the power requirements also which are shown in different graphs. These graphs are understood by the software developed and recognized for the words. Moreover, the program has the flexibility to be modified.

By this approach we can read text from a document, Web page or e-Book and can generate synthesized speech through a computer's speakers. This can save time by allowing the user to listen background materials while performing other tasks. This approach can be use in part as well. If we want only image to text conversion then it can be possible and if we want only text to speech conversion then it is possible easily. People with poor vision or visual dyslexia or totally blindness can use this approach for reading the documents and books easily. People with speech loss or totally dumb person can utilize this approach to turn typed words into vocalization. But now it is not only serving that purpose but certain hi-tech applications are also using this methodology where the men can not physically go.

7.2 Future scope

An image to speech conversion system is developed still work can be done in this field. Since printed document images archived by many applications are more of historical and poor in quality, there is a need to apply advanced image pre-processing techniques for document analysis. Document image processing algorithms for document image collections need more progress. Schemes that learn from document image collections itself for better performance are needed. Recognition from poor quality documents results in a number of recognition errors. Retrieval of documents in this situation requires more functionality. Effective schemes for retrieval in presence of OCR errors there is need to develop Multilingual OCR system so that we can read more than one language documents. The facility to stop, pause and continue reading can be provided, i.e., the user should be able to pause the synthesizer at any time and then continue reading using just a mouse-click. Accent variation and multiple voices (both male and female versions) can be provided to users to choose depending upon their interest.

References

[1] Ainsworth, W., "A system for converting English text into speech," Audio and Electroacoustics, IEEE Transactions on , vol.21, no.3, pp. 288-290, Jun 1973

[2] Fushikida, Katsunobu; Mitome, Yukio; Inoue, Yuji, "A Text to Speech Synthesizer for the Personal Computer," Consumer Electronics, IEEE Transactions on , vol.CE-28, no.3, pp.250-256, Aug. 1982

[3] Hertz, S., "English text to speech conversion with delta," Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86. , vol.11, no., pp. 2427-2430, Apr 1986

[4] Lynch, M.R.; Rayner, P.J., "Optical character recognition using a new connectionist model," Image Processing and its Applications, 1989., Third International Conference on , vol., no., pp.63-67, 18-20 Jul 1989

[5] Malyan, R.R.; Sunthankar, S.; Teranchi, H.; Yeghiazarian, A., "Perception of multi-author handprinted text," Character Recognition and Applications, IEE Colloquium on , vol., no., pp.8/1-8/3, 2 Oct 1989

[6] Curtis, K.M.; Race, P.; Aziz, A.A., "Parallelism and the transputer in the automatic translation of text to speech," Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on , vol., no., pp.809-811 vol.2, 23-26 May 1989

[7] Mori, S.; Suen, C.Y.; Yamamoto, K., "Historical review of OCR research and development," Proceedings of the IEEE , vol.80, no.7, pp.1029-1058, Jul 1992

[8] Jianli Liu; Nugent, J.H.; Bowen, D.G.; Bowen, J.E., "Intelligent OCR editor," Electrical and Computer Engineering, 1993. Canadian Conference on , vol., no., pp.9-11 vol.1, 14-17 Sep 1993

[9] Elliman, D.G., "Peeling potatoes with a cheese grater [handwritten document OCR] ," Handwriting Analysis and Recognition: A European Perspective, IEE European Workshop on , vol., no., pp.14/1-14/4, 12-13 Jul 1994

[10] Blando, L.R.; Kanai, J.; Nartker, T.A., "Prediction of OCR accuracy using simple image features," Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on , vol.1, no., pp.319-322 vol.1, 14-16 Aug 1995

[11] Lucas, S.M., "High performance OCR with syntactic neural networks," Artificial Neural Networks, 1995., Fourth International Conference on , vol., no., pp.133-138, 26-28 Jun 1995

[12] Katae, N.; Kimura, S., "Natural prosody generation for domain specific text-to-speech systems," Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on , vol.3, no., pp.1852-1855 vol.3, 3-6 Oct 1996

[13] Leija, L.; Santiago, S.; Alvarado, C., "A system of text reading and translation to voice for blind persons ," Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the IEEE , vol.1, no., pp.405-406 vol.1, 31 Oct-3 Nov 1996

[14] Chen Fang; Yuan Baozong, "Intelligent speech production system with text generation," Signal Processing, 1996., 3rd International Conference on , vol.1, no., pp.769-772 vol.1, 14-18 Oct 1996

[15] Tatham, M.; Lewis, E., "Improving text-to-speech synthesis," Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on , vol.3, no., pp.1856-1859 vol.3, 3-6 Oct 1996

[16] Tanprasert, C.; Koanantakool, T., "Thai OCR: a neural network application," TENCON '96. Proceedings. 1996 IEEE TENCON. Digital Signal Processing Applications , vol.1, no., pp.90-95 vol.1, 26-29 Nov 1996

[17] Vergin, R.; O'Shaughnessy, D.; Farhat, A., "Time domain technique for pitch modification and robust voice transformation," Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on , vol.2, no., pp.947-950 vol.2, 21-24 Apr 1997

[18] Breen, A.P., "The future role of text to speech synthesis in automated services ," Advances in Interactive Voice Technologies for Telecommunication Services (Digest No: 1997/147), IEE Colloquium on , vol., no., pp.6/1-6/5, 12 Jun 1997

[19] Leija, L.; Hernandez, P.; Santiago, S., "Reader instrument of basic texts to the teaching of blind people ," [Engineering in Medicine and Biology, 1999. 21st Annual Conf. and the 1999 Annual Fall Meeting of the Biomedical Engineering Soc.] BMES/EMBS Conference, 1999. Proceedings of the First Joint , vol.1, no., pp.588 vol.1-, 1999

[20] Bazzi, I.; Schwartz, R.; Makhoul, J., "An omnifont open-vocabulary OCR system for English and Arabic," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.21, no.6, pp.495-504, Jun 1999

[21] Sun-Hwa Hahn; Joon Ho Lee; Jin-Hyung Kim, "A study on utilizing OCR technology in building text database," Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on , vol., no., pp.582-586, 1999

[22] Jaehwa Park; Govindaraju, V.; Srihari, S.N., "OCR in a hierarchical feature space," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.22, no.4, pp.400-407, Apr 2000

[23] Ishitani, Y., "Model-based information extraction method tolerant of OCR errors for document images," Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on , vol., no., pp.908-915, 2001

[24] Lecoq, J.C.; Najman, L.; Gibot, O.; Trupin, E., "Benchmarking commercial OCR engines for technical drawings indexing ," Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on , vol., no., pp.138-142, 2001

[25] Desrochers, D.; Qu, Z.; Saengdeejing, A., "OCR readability study and algorithms for testing partially damaged characters," Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on , vol., no., pp.397-400, 2001

[26] Ghayoori, A.; Hendessi, F.; Sheikh, A., "Smooth ergodic hidden Markov model and its applications in text to speech systems," Intelligent Multimedia, Video and Speech Processing, 2004. Proceedings of 2004 International Symposium on , vol., no., pp. 234-237, 20-22 Oct. 2004

[27] Nagy, G.; Prateek Sarkar, "Document style census for OCR," Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on , vol., no., pp. 134-147, 2004

[28] Sarfraz, M.; Zidouri, A.; Shahab, S.A., "A novel approach for skew estimation of document images in OCR system," Computer Graphics, Imaging and Vision: New Trends, 2005. International Conference on , vol., no., pp. 175-180, 26-29 July 2005

[29] Shaolei Feng; Manmatha, R., "A hierarchical, HMM-based automatic evaluation of OCR accuracy for a digital library of books," Digital Libraries, 2006. JCDL '06.

Proceedings of the 6th ACM/IEEE-CS Joint Conference on , vol., no., pp.109-118, June 2006

[30] Goto, H., "OCRGrid : A Platform for Distributed and Cooperative OCR Systems," Pattern Recognition, 2006. ICPR 2006. 18th International Conference on , vol.2, no., pp.982-985, 2006

[31] Dey, S.; Kedia, M.; Basu, A., "Architectural Optimizations for Text to Speech Synthesis in Embedded Systems," Design Automation Conference, 2007. ASP-DAC '07. Asia and South Pacific , vol., no., pp.298-303, 23-26 Jan. 2007

[32] Bitouk, D.; Nayar, S.K., "Creating a Speech Enabled Avatar from a Single Photograph," Virtual Reality Conference, 2008. VR '08. IEEE , vol., no., pp.107-110, 8-12 March 2008